Edge Detection with Bilateral Filtering in Spiral Space

Jun Zhou, Qimei Hu and Xiangjian He

Abstract-- **Edge detection is a vital preprocessing step towards high-level image analysis. One of its many applications of edge is that it can be used in image compression where accurate edge detection is required. A way for improving the accuracy and quality of edge detection of noisy contaminated image is to preserve edge details while removing noise. In this paper Spiral Architecture is used to sample image data. Spiral Architecture provides powerful computational power and enables image to be uniformly partitioned and distributed to various processors for parallel processing. This paper shows the implementation of recently developed bilateral filtering technique in Spiral Architecture for edge-preserving smoothing of noise in images.**

***Index Terms*—Image compression, Spiral Architecture, Edge detection.**

I. INTRODUCTION

Image processing is a vital cost-effective technology in many fields today including astronomy, medicine, crime, remote sensing, manufacturing, entertainment and multimedia. Edge detection is a process of detecting areas of abrupt changes or discontinuities in some visual property (light intensity, texture or colour). It is a critical preprocessing step towards high-level image understanding. Firstly because edges are essentially surface boundary discontinuities they hold important structural information about objects in an image (e.g. size, shape and location) that subsequent processing highly depends on. Secondly it is very difficult to recover from errors made at this stage [1].

A. *Differential Edge Detection*

The differential detection method is the most commonly used approach to edge detection and can be further divided into two classes: first- and second-order derivative edge detection. For the first-order class, a pixel is an edge point if the gradient magnitude at this pixel is greater than a threshold value. For the second-order derivative class, an edge is where the derivative evaluates to zero.

Differential edge detection method uses approximation of spatial gradient at each pixel location. Denote $f(x, y)$ to be the function that maps gray scale value at a particular pixel a_0 to its Cartesian co-ordinates. Let $G(x, y)$ be the rate of change in gray scale value at pixel a_0. $G(x, y)$ can be computed in terms of the derivatives along x and y directions, $G_x(x, y)$ and $G_y(x, y)$ as follows:

$$G(x, y) = \{ [G_x(x, y)]^2 + [G_y(x, y)]^2 \}^{\frac{1}{2}} \quad (1)$$

The technique for approximation of derivatives $G_x(x, y)$ and $G_y(x, y)$ is through a mathematical operation termed *convolution*. In image processing convolution involves the image array and a smaller array called *kernel* or *mask*. As the mask moves over the image array, calculations are performed for the centre pixel of each local area (neighbourhood) underlying the mask.

Differentiation is a noise enhancing operation and the higher the order of derivative the more pronounced the effect. Therefore differentiation often has to be preceded by smoothing for images corrupted by noise.

B. *Smoothing*

Smoothing reduces the sharpness of transitions in intensity values to achieve noise reduction or detail suppression. Two approaches to smoothing are linear and nonlinear.

Linear filtering is implemented by convolution of the original image function with a predefined kernel or mask. In the past decade nonlinear filters have been developed to achieve a more desirable level of smoothing in applications where important visual cues provided by edges need to be preserved and less blurry effect introduced than linear filters. Many efforts have been devoted to *edge-preserving smoothing* [2]-[4].

In 1998, **bilateral** filtering was introduced [5]. In essence a bilateral filter replaces a given pixel value with an average of similar and nearby pixel intensity values. In this form of filtering, a range filter is combined with a domain filter. Domain filtering enforces spatial closeness by weighing pixel values with coefficients that fall off with distance. A range filter, on the other hand, assigns greater coefficients to those neighbouring pixels with light intensity that is more similar to the centre pixel value. Hence the original intensity value at a given pixel would be better preserved thanks to range filtering. Range filtering by itself is little use because pixel values that are far away from a given pixel should not contribute to the new value.

The kernel coefficients of a bilateral filter are determined by the combined closeness and similarity function.

Let $f : \Re^2 \rightarrow \Re$ be the original brightness function of an image which maps the coordinates of a pixel (x, y) to a value in light intensity. Then for any given pixel a at (x, y) within a

All the authors are with the Computer Vision Research Group, University of Technology, Sydney, PO Box 123, Broadway, NSW 2007, Australia.

ICITA2004 ISBN 0-646-42313-4

neighbourhood of size n, which has a_0 as its centre, its coefficient assigned by the range filter $r(a)$ is determined by the similarity function s:

$$r(a) = s(f(a), f(a_0)) \qquad (2)$$

Similarly, its coefficient assigned by the domain filter $g(a)$ is determined by the closeness function c:

$$g(a) = c(a, a_0) \qquad (3)$$

Then for the central pixel of the neighbourhood a_0, its new value, denoted by $h(a_0)$,

$$h(a_0) = k^{-1} \sum_{i=0}^{n-1} f(a_i) \times g(a_i) \times r(a_i) \qquad (4)$$

k is the normalization constant and is defined as follows

$$k = \sum_{i=0}^{n-1} g(a_i) \times r(a_i) \qquad (5)$$

The normalizer k is necessary because the average image intensity should not be affected by multiplying the mask with the original image.

The challenge to precisely detect edges in real images has inspired a variety of advanced edge detection algorithms. Multi-scale analysis is a systematic approach to edge detection. This approach relies on the inspection of intensity changes on different scales.

C. *Multi-scale edge detection*

Objects in the world only exist as meaningful entities over a limited range of scale. Hence the physical description strongly depends on the scale. When analyzing measured data such as images, without any prior knowledge, there is no reason to favour any particular scale. The idea is that for any image, a set of gradually smoothed or simplified images should be generated, in which fine scale structures are successively suppressed.

A formal definition for continuous signals of arbitrary dimensions N was given by Lindeberg [6]. The idea is that multi-scale representation of a measured signal could be obtained by embedding the signal in a one-parameter family of derived signals. The parameter represents the level of scale. By successively increasing the scale t, details will be gradually smoothed out.

D. *Spiral Architecture*

In this research project, the image arrays are mapped into Spiral Architecture, which is a recently developed sampling technique. Research work by Sheridan [7] has shown that *Spiral Architecture* and associated processing paradigm offers several advantages over the two-dimensional array representation. Each element (hexagon) in the Spiral Architecture has only six neighbouring hexagons. This feature implies less memory space and computation required for processing. Spiral Architecture also enables the traditional two-dimensional visual field to be represented as a one-dimensional array. Figure 1 displays such architecture with a collection of $7 \times 7 = 49$ hexagons. Each cell is uniquely addressed and the spiral addresses grow from the centre clockwise in base of seven.

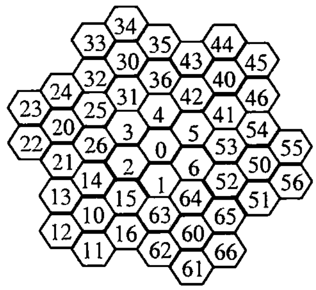


Figure 1. Spiral Architecture with 49 hexagons.

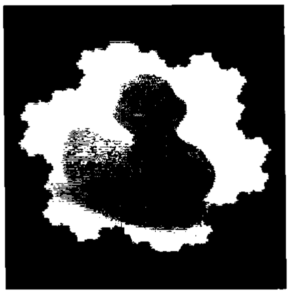


Figure 2. Sample image "duck" in Spiral space

Because currently there is no supporting hardware for Spiral Architectural sampling, a mimic system was introduced by He [8]. The mimic system retains the organizational character of the original Spiral Architecture. Hence it inherits the computational power of the original structure. In the mimic system, one hexagonal pixel is formed by four neighbouring pixels in the traditional system and its gray scale value is the average of those four pixels values. Figure 2 is a sample image represented in mimic Spiral Architecture.

II. MULTI-SCALE EDGE DETECTION WITH BILATERAL FILTERING IN SPIRAL ARCHITECTURE

A. *Bilateral filtering in Spiral Space*

In this research, edge detection is accomplished by applying a new bilateral filtering technique specifically designed for Spiral Architecture, integrating the multi-scale approach to edge detection.

The domain filter in Gaussian form with scale value of t is defined as follows:

$$g(x, y; t) = e^{-\frac{x^2+y^2}{2t}} \qquad (6)$$

The range filter kernel coefficients for these 7 mimic hexagons will be generated from the following function:

$$r(a_i) = e^{-\frac{|f(a_i)-f(a_0)|^2}{2\sigma_r^2}}$$

Application of the new bilateral smoothing filter produces, for each pixel in the image, a weighted average such that the central pixel a_0 contribute more significantly to the result than its neighbouring pixels. Pixels with more similar intensity value or closer to the central pixel contribute more than those with more different value or further away. Level of smoothness depends on the size of the neighbourhood chosen and the geometric spread of light intensity from the central pixel value. The bigger the size and spread, the higher level of smoothness will be achieved.

B. Edge detection in Spiral Architecture

Let $L_x(x, y;t)$ and $L_y(x, y;t)$ be the derivative of $L\ (x, y;t)$ with respect to the x and y direction at a_0 for a given scale *t*. Denote gradient magnitude $L_i = \sqrt{L_x^2 + L_y^2}$ at $a_0, a_1, a_2, a_3, a_4, a_5, a_6$ by $L_0, L_1, L_2, L_3,\ L_4,\ L_5, L_6$ respectively. L_r, is the average light intensity at the middle point between a_5 and a_6. L_l is the average of light intensity at points a_2 and a_3. Note that the distance between a_l and a_0, a_r and a_0, a_1 and a_0, and between a_4 and a_0 is 2. Then we have approximation of gradient magnitude along x and y direction

$$L_x(x, y;t) = \frac{\frac{L_r - L_0}{2} + \frac{L_0 - L_l}{2}}{2}$$

$$= \frac{1}{4}(L_r - L_l)$$

$$= \frac{1}{4}\{\frac{1}{2}[L(x+2, y-1;t) + L(x+2, y+1;t)] - \frac{1}{2}[L(x-2, y-1;t) + L(x-2, y+1;t)]\}$$

and similarly

$$L_y(x, y;t) = \frac{(L_4 - L_0) + (L_0 - L_1)}{4}$$

$$= \frac{1}{4}(L_4 - L_1)$$

$$= \frac{1}{4}[L(x, y-2) - L(x, y+2)]$$

In discrete space, zero-crossings of 2nd derivatives of *L* do not always lie at the sample pixel locations. This research uses He's proposed procedure based on first order derivative and its orientation with respect to the x direction to determine the presence of an edge. Let α be the angle from the vector (1,0) to the vector (L_x, L_y) (see figure 3). The following method is used to judge the edge points.

Gradient Direction

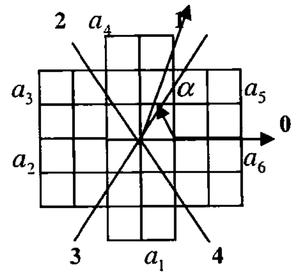


Figure 3. Gradient direction α with respect to the x direction.

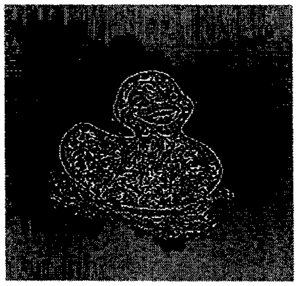
Let $A = tg(\alpha)$, then $A = \frac{L_y(x, y;t)}{L_x(x, y;t)}$

When L_x and L_y are both equal to zero, α is set to zero; and if L_x is zero while L_y is not, α is 90 degrees. Otherwise:

- If α falls into the area between line 0 and 1or line 0 and 3, it means that $L\ (x, y;t)$ is most significantly affected by a_2 or a_5
  — if ($L_0 \geq L_2$ and $L_0 > L_5$) or ($L_0 > L_2$ and $L_0 \geq L_5$)
  then a_0 is an edge point
- If α falls into the area between line 1 and 2 or line 3 and 4, it means that $L\ (x, y;t)$ is most significantly affected by a_1 or a_4
  — if ($L_0 \geq L_1$ and $L_0 > L_4$) or ($L_0 > L_1$ and $L_0 \geq L_4$)
  then a_0 is an edge point.
- If α falls into the area between line 2 and 0 or line 4 and 0, it means that $L\ (x, y;t)$ is most significantly affected by a_3 or a_6
  — if ($L_0 \geq L_3$ and $L_0 > L_6$) or ($L_0 > L_3$ and $L_0 \geq L_6$)
  then a_0 is an edge point.

III. EXPERIMENTAL RESULTS

Shown here are results for the "duck" image. Figure 4 (a) is the initial edge map of the original image with 2858 edge points. After bilateral smoothing with kernel size of 7, the new edge map in (b) has 850 less edge points. When the kernel size increased to 19, the number of edge points decreases to 1484 points. As shown in the edge maps, the outlines of the object and some fine features such as the spots in duck's eyes have been retained and not lost as result of smoothing.



(a) Initial edge map

(b) After smoothing with kernel size of 7

(c) After smoothing with kernel size of 19

Figure 4. Edge detection results with multi-scale bilateral smoothing in Spiral Architecture

REFERENCES

[1] Bräunl, T., S. Feyrer, W. Rapf and M. Reinhardt. *Parallel Image Processing*. Springer, Berlin, 2001.

[2] Kotropoulos, C., and Pitas, I. (ed.), *Nonlinear Model-Based Image/Video Processing and Analysis*. John Wiley & Sons, Inc. New York, 2001.

[3] R. T. Chin and C. L. Yeh. *Quantitative evaluation of some edge-preserving noise-smoothing techniques*. CVGIP, 23: 67-91, 1983.

[4] M. Nagao and T. Matsuyama. *Edge preserving smoothing*. CGIP, 9: 394-407, 1979.

[5] C. Tomasi and R. Manduchi, *Bilateral Filtering for Gray and Color Images*, Proceedings of the 1998 IEEE international Conference on Computer Vision, Bombay, India.

[6] Lindeberg, T., *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, London, 1994.

[7] Sheridan, P., *Spiral Architecture for machine vision*. Ph.D Thesis, school of Computing Sciences, University of Technology, Sydney, Australia, 1996.

[8] He, X., *2D Object Recognition with Spiral Architecture*. PhD thesis. University of Technology, Sydney, Australia, 1999.