BI-LATERAL FILTERING BASED EDGE DETECTION ON HEXAGONAL ARCHITECTURE

Qiang Wu, Xiangjian He, Tom Hintz

Department of Computer Systems
University of Technology, Sydney, Australia
Email: {wuq, sean, hintz}@it.uts.edu.au

ABSTRACT

Edge detection plays an important role in image processing area but is still an open problem. This paper presents a novel edge detection method based on bi-lateral filtering which achieves better performance than single Gaussian filtering. In this form of filtering, both spatial closeness and intensity similarity of pixels are considered in order to preserve important visual cues provided by edges and reduce the sharpness of transitions in intensity values as well. In addition, the edge detection method proposed in this paper is achieved on hexagonally sampled images. Due to the compact and circular nature of the hexagonal lattice, a better quality edge map is obtained on hexagonal architecture than common edge detection on square architecture. Experimental results using our proposed method in this paper exhibit encouraging performance.

1. INTRODUCTION

In general, images contains enormous amount of data. Often, only part of this information is necessary to solve a specific computer imaging problem. So a proper pre-processing is required to isolate the essential information in order to simplify the subsequent work. One such processing that is commonly employed is edge detection. As the success of subsequent processing is sensitive to the quality of edge detection, the performance of higher level processes rely heavily on the complete and correct determination of edges.

During the last three decades, many algorithms have been developed for edge detection e.g. Roberts edge detector, Sobel edge detector, Marr-Hildreth edge detector, and etc. All these edge detectors resolved some problems but still have their disadvantages. In 1986, Canny [1] developed a optimal edge detection which is widely accepted as a good edge detection in terms of good detection, good localization, and single response. Canny edge detection uses linear filtering with a Gaussian kernel to suppress noise and reduce the sharpness of transition in intensity values. Then, candidate edge points are detected using local non-minimal suppression in the smoothed images. In order to recover missing weak edge points and eliminate false edge points, two edge strength thresholds are set to examine all the candidate edge points. Those below the lower threshold are marked as non-edge. Those which are above the lower threshold and can be connected to a point whose edge strength is above the higher threshold through a chain of edge points are marked as edge points. Canny edge detection is able to extract not only step edges but also ridge and roof edges.

However, the performance of Canny edge detection relies on Gaussian filtering. Gaussian filtering not only removes image noise, suppresses image details but also weakens the edge information. Consequently, raising threshold in Canny edge detection will lose edge points and decreasing threshold will introduce more false edge points. In this paper, an additional filtering called range filtering [2] is integrated into common Gaussian filtering in order to reduce the blur effect of smoothing on potential edge points. Such so-called bi-lateral filtering preserves the major information provided by edge points during smoothing processes, and removes image noise and trivial details.

Moreover, the success of Canny edge detection is often limited when it comes to curved features. This is primarily due to the algorithm's effectiveness being affected by the quality of input image based on square architecture. Although using high resolution square grids is able to represent the data with sufficient fidelity, it increases the computational complexity. Alternatively, the proposed edge detection in this paper is implemented on hexagonal architecture. Hexagonal lattice promises better efficiency and less aliasing [3]. In addition, it also makes humans less sensitive to edges in the diagonal rather than horizontal or vertical direction[4].

The organization of this paper is as follows. Hexagonal image architecture is introduced in Section 2 followed by the explanation on bi-lateral filtering in Section 3. Section 4 presents the experimental results. Conclusions are given in Section 5.

0-7803-8874-7/05/$20.00 ©2005 IEEE

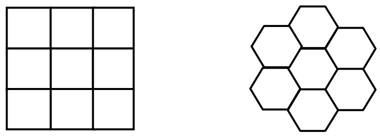


Fig.1. Two different image architectures

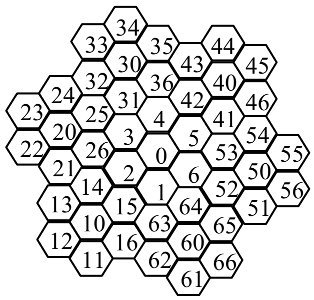


Fig. 2. 49 hexagonal pixels labeled by spiral addresses

2. HEXAGONAL IMAGE ARCHITECTURE

The possibility of using a hexagonal grid to represent digit images has been studied for more than thirty years. Hexagonal grids have higher degrees of symmetry than the square grids. This symmetry results in a considerable saving of both storage and computation time [3, 5]. On hexagonal architecture, each grid has only six neighboring grids which have the same distance to the centre hexagon of the seven-hexagon unit. It is a different grid arrangement scheme from square architecture which uses a set of 3×3 vision unit (see Fig. 1.). Because of the uniformly connected and close-packed form, greater angular resolution, higher efficiency, and better performance are achieved in many hexagonal systems [3, 5].

In order to widely utilize hexagonal grid system on image processing area, many mathematic foundations have been specially developed for hexagonal grid. Her [5] proposed a symmetrical hexagonal coordinate frame using integer coordinate of three components to identify each grid on hexagonal architecture. However, Her still borrowed mathematics from the square system to apply his hexagonal coordinates frame on image transformation. Moreover, because one more component was introduced in his coordinate system than traditional square architecture with two components, it increased computational complexities in some degree. Sheridan [6] proposed another single component hexagonal addressing system called spiral addressing. On Spiral Architecture, each pixel is identified by a designated positive number. The numbered hexagons form the cluster of size 7^n. The hexagons tile the plane in a recursive modular manner along the spiral direction (see Fig. 2.). Moreover, Sheridan [6] also developed two algebraic operations on Spiral Architecture: Spiral Addition and Spiral Multiplication. The neighboring relation among the pixels on Spiral Architecture can be expressed uniquely by these two operations. These two operations also define two transformations on spiral address space respectively, which are image translation and image rotation.

In spite of the many advantages of hexagonal architecture, it has not been used widely in image processing area. The main reason is that there are no mature devices for capturing image and for displaying image based on hexagonal architecture. In 2004, Wu [7] proposed virtual hexagonal architecture which successfully achieved image mapping between square architecture and hexagonal architecture almost without distortion. Moreover, such virtual hexagonal architecture retains the symmetrical properties of hexagonal grid system. It almost does not change the resolution of the original image during mapping procedure. In this paper, the proposed algorithm is implemented on virtual hexagonal architecture.

Sobel edge detector [8] is a gradient based edge detector. It is used widely in image processing area as a good 3×3 operator. In order to implement edge detection on hexagonal architecture, a modified Sobel operator is presented as in Fig. 3. Using the modified Sobel operator, edge length and direction can be calculated on hexagonal architecture in the similar way as on square architecture.

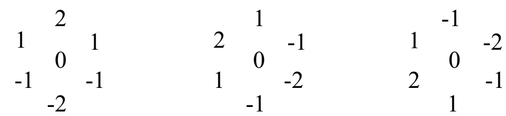


Fig. 3. Modified Sobel operator

3. BI-LATERAL FILTERING

Before the edge map of an image is found, it is common that image noise is removed (or suppressed) by applying a filter that blurs or smooths the image. Smoothing reduces the sharpness of transitions in intensity values to achieve noise reduction or detail suppression. Two approaches for 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. Bilateral filtering [2] is one such technique that takes into account intensity similarity in order to reduce the blurry effect of smoothing on the potential edge points. 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 pixel values that are 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 of 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 $f:\Re^2 \to \Re$ be the original brightness function of an image which maps the coordinates of a pixel, (x, y) to a value in light intensity. Let a_0 be the reference pixel. Then, for any given pixel a at (x, y) with the image size n, its coefficient assigned by the range filter $r(a)$ is determined by the similarity function s:

$$r(a) = s(f(a), f(a_0)) = e^{-\frac{(f(a)-f(a_0))^2}{2\sigma_1^2}} \quad (1)$$

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

$$g(a) = c(a, a_0) = e^{-\frac{(a-a_0)^2}{2\sigma_2^2}} \quad (2)$$

That is for the reference pixel a_0, its new value, denoted by $h(a_0)$, is

$$h(a_0) = k^{-1}\sum_{i=0}^{n-1} f(a_i)\times g(a_i)\times r(a_i) \quad (3)$$

where k is the normalization constant and is defined as

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

Equation (3) above is called a convolution of the image brightness function $f(\bullet)$ with domain filter g and range filter r. The normalized k is necessary because the average image intensity should not be affected by multiplying the mask with the original image.

Choosing Gaussian function to be closeness function and similarity function (Equation 1 and 2) can simplify the practical processing. σ_1 and σ_2 are standard deviation in Gaussian function. In Equation (3), n is the size of input image. For a large image, it will take a longer time to carry on convolution processing. Considering about 90% energy concentrates in the central area of "Mexico cap" (the curve of Gaussian function) in the width of $2\times\sigma$, Equation (3) can be carried on in a smaller size window as long as it can cover the reference pixel's neighboring area of $2\times\sigma$.

Fig. 4. Original image, Lena

4. EXPERIMENTAL RESULTS

To study the effect of new edge detection method on hexagonal architecture, 8-bit grey level Lena image is chosen as the image to be processed, because it is one of most widely accepted testing image to evaluate the performance of edge detection algorithm (see Fig. 4.).

Three different edge maps (see Fig. 6) are produced in order to demonstrate the performance improved by new edge detection methods. The first edge map is produced by common Canny edge detection on square architecture. The second edge map is produced from virtual hexagonal architecture. Here, the input image is mapped to virtual hexagonal architecture first. Then, the pixels' gradient values are calculated using the modified Sobel operator in Fig. 3 after common Gaussian filtering. Similar method as Canny edge detection is carried on hexagonal architecture to find the edge points. Namely, candidate edge points are located first using local non-minimal suppression. Exact edge points are filtered out using double thresholds following the same approach adopted by Canny method. The third edge map is produced using the same method to generate the second edge map except that the input image is pre-processed by bi-lateral filtering rather than single Gaussian filtering.

All the above edge maps are generated based on the consistent environment parameters. For Gaussian processing including domain filtering, $\sigma = 3.01$. For range filtering, σ_1 equals to the standard deviation of grey values in input image. The size of window for carrying on convolution processing in Equation (3) is 49. That means, for square architecture, this window is a 7×7 block. For hexagonal architecture, this window is a cluster of 49 hexagonal lattices as shown in Fig. 2. In addition, the same double thresholds are used to locate the exact edge points while producing all edge maps. The higher threshold is 0.07 and the lower threshold is 0.015.

Fig. 5. shows three images processed by different filtering operations on different image architecture. It is shown that changing image architecture does not affect Gaussian processing. However, compared with Gaussian processing, bi-lateral filtering sharpens the major edge area information as well as smoothes other pixels like Gaussian filtering.

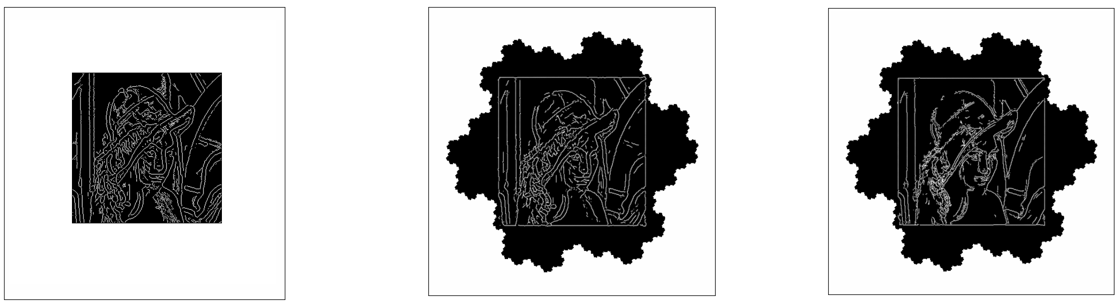
Fig. 6. gives three edge maps which are produced after different filtering processing. It is obvious that, after

(a) Gaussian filtering on square architecture

(b) Gaussian filtering on hexagonal architecture

(c) Bi-lateral filtering on hexagonal architecture

Fig. 5. Images processed by different filtering operations on different image architectures



(a) Edge map after Gaussian filtering on square architecture

(b) Edge map after Gaussian filtering on hexagonal architecture

(c) Edge map after bi-lateral filtering on hexagonal architecture

Fig. 6. Edge maps detected after different filtering operations on different image architectures

Gaussian filtering, edge map obtained on hexagonal architecture has more clear background than the one obtained on square architecture without losing major edge points. This is mainly due to the uniformly connected and close-packed structure of hexagonal lattice. Moreover, one more mask introduced in edge detection operator (see Fig. 3.) than common operator with only two masks on square architecture improves accuracy while detecting edge points. In addition, Fig. 6 shows that besides advantages provided by hexagonal architecture, bi-lateral filtering also improves the edge map quality, because it enhances edge area information and suppresses other image noise and trivial edge pixels as well before detecting edge points.

5. CONCLUSIONS

In this paper, a novel edge detection method is presented. The use of bi-lateral filtering combined with the advantages of hexagonal image architecture has achieved encouraging edge detection performance under the same experimental conditions. It is also found that sigma values in Equation (1) and (2) are still the significant parameters which will affect the performance of algorithm.

6. REFERENCES

[1] J. F. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. PAMI-8, pp. 679-698, Nov. 1986.

[2] D. Barash, "Fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 844-847, 2002.

[3] R. M. Mersereau, "The Processing of Hexagonally Sampled Two-Dimensional Signals" *Proc. the IEEE*, Vol. 67, pp. 930-949, 1979.

[4] R. M. Gray, P. C. Cosman, and K. L. Oehler, *Digital Images and Human Vision*. Cambridge, MA: MIT Press, 1993.

[5] I. Her, "Geometric Transformations on the Hexagonal Grid," *IEEE Trans. on Image Processing*, vol. 4, 1995.

[6] P. Sheridan, T. Hintz, and D. Alexander, "Pseudo-invariant Image Transformations on a Hexagonal Lattice," *Image and Vision Computing*, vol. 18, pp. 907-917, 2000.

[7] Q. Wu, X. He, and T. Hintz, "Virtual Spiral Architecture" *Proc. International Conference on Parallel and Distributed Processing Techniques and Applications*, Las Vegas, Nevada, USA, Vol. 1, pp. 399-405, 2004.

[8] L. S. Davis, "A Survey of Edge Detection Techniques," *Computer Graphics and Image processing*, vol. 4, pp. 248-270, 1975.