Edge Map Improvement on Spiral Architecture

Qiang Wu, Xiangjian He and Tom Hintz
Department of Computer Systems, University of Technology, Sydney
PO Box 123, Broadway 2007, Australia
{wuq, sean, hintz}@it.uts.edu.au

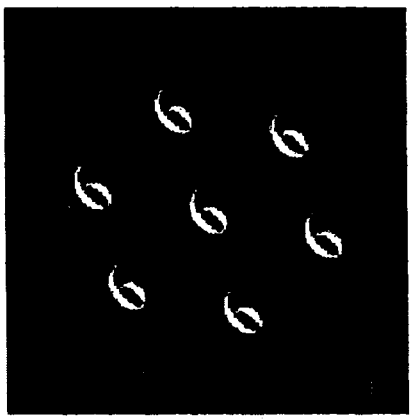
Abstract *Edge map is considered as an important entity containing most of object features in an image. Many computer vision systems rely on the use of the boundary line information to perform the object recognition tasks. However, with the exception of images acquired from highly restricted environment, common edge detectors do not guarantee the production of continuous boundaries of objects. In this paper, a local processing based edge-linking algorithm is proposed. We set a criterion involving multiple properties of the pixel to find the best candidate points for edge linking. In addition, this algorithm is implemented on Spiral Architecture.*

Keywords: edge detection, edge restoration, Spiral Architecture, image processing

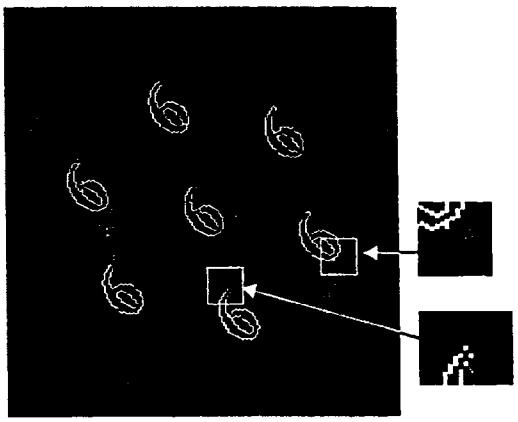
1 Introduction

Edge maps are considered as one of the most important features of object in an image. We are able to gain insight into the entire image analysis by only studying edge map. Many computer vision systems rely on the use of the boundary line information to derive the object shapes, to measure the object parameters and to perform the object recognition tasks.

However, with the exception of images acquired from highly restricted environment, common edge detectors do not guarantee the production of continuous boundaries of objects [1-5]. Very often, the result of edge detection is a collection of edges from separated segments with many gaps. Moreover, such edges often do not feature single pixel width feature (see the following further explanation). Figure 1 gives a result of edge detection by the gradient based approach. Zoomed areas highlight the broken points and the 'thick' edge segments. Computer vision systems based on such edges have difficulties for successfully performing their given task. So an edge improvement



a) Original Image



b) Edge Map

Figure 1 The result of edge detection on Spiral Architecture

procedure is required after edge detection operation. This process attempts to fill in the gaps along the edges and to link the broken edge segments such that the precise description and accurate analysis of object boundaries are feasible.

Among many categories of edge-linking methods, one is called the category of global processing methods, which applies mathematical modelling techniques to formulate the boundaries of objects. Hough transformation [6] is a typical example of global processing. The Hough transformation maps the edge elements in an image space to a parametric space. The parameters used to make mathematical equations describing the object boundaries can be extracted from the second space. Then, the obtained mathematical curves are used to improve the original edge map. Most of global processing methods are suffered with the computational complexity. Moreover, many of these methods have fine properties for objects in certain application areas, but not for use on general images containing edges of the arbitrary shapes [7].

Canny [1] proposed methods using thresholds for edge-linking. He set two thresholds to measure all the points in the gradient image. Initial edge map is produced by the points whose gradients are above the higher threshold. Another edge map involving more points above the lower threshold is used as the reference map to fill in the gaps. Threshold method belongs to local processing category, which attempts to close the edge gaps by seeking the edge pixels which are most likely connected in the neighbourhood of an edge broken point. Compared with global processing method, local processing method features its light computation and saves the complex task of building mathematical models.

In this paper, another local processing based edge-linking algorithm is proposed. We set a criterion involving multiple properties of the pixel to determine the best candidate points for edge linking. This algorithm consists of three steps, edge thinning, edge linking and cracks elimination. This algorithm is implemented on Spiral Architecture, which is made up of hexagonal pixels rather than rectangle pixels. In general, Spiral Architecture will provide the higher computation accuracy to the local

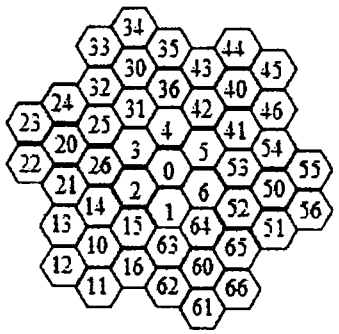


Figure 2 Cluster of size 49 including spiral addresses

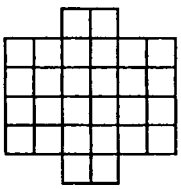


Figure 3 Mimic Spiral Architecture

processing using the information in a neighbouring area than the rectangle architecture, because the distances between each point and its six neighbouring points are same on Spiral Architecture.

The organization of this paper is as follows. A brief introduction about Spiral Architecture is shown in Section 2. Section 3 shows the edge improvement procedure and contains the incremental experimental results. We conclude in Section 4.

2 Spiral Architecture

Spiral Architecture is made up of the hexagonal pixels arranged in a spiral clusters. This cluster consists of the organizational units of seven-hexagon compared with the traditional rectangular image architecture using a set of 3×3 vision unit. In the Spiral Architecture, any pixel has only six neighboring pixels which have the same distance to the centre hexagon of the seven-hexagon unit of vision. 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. An example of a cluster with size of 7^2 and the corresponding addresses are shown in Figure 2.

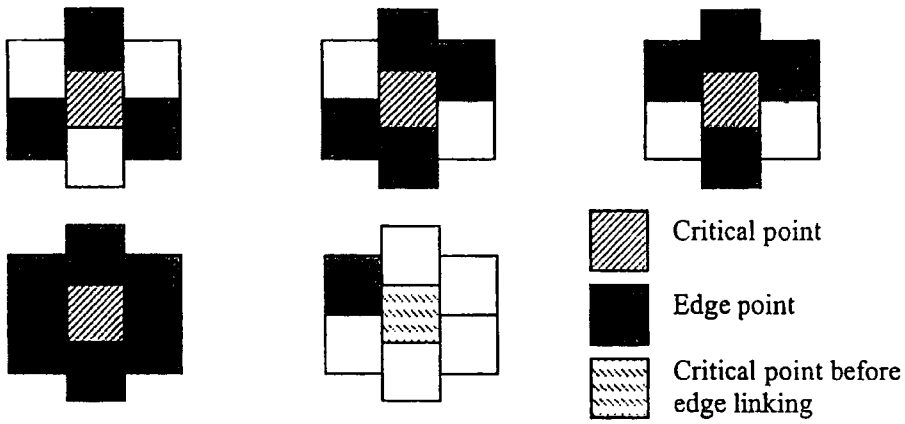


Figure 4 Critical point in edge map

Currently, since we cannot obtain a real image display device based on Spiral Architecture, we have to use mimic Spiral Architecture for research purpose, which uses four rectangle pixels to mimic one hexagonal pixel (See Figure 3).

Spiral Architecture contains very useful geometric and algebraic properties, which can be interpreted in terms of the mathematical object, Euclidean ring (refer to [8] for details). Two algebraic operations have been defined 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 mentioned above define two transformations on spiral address space respectively, which are image translation and image partitioning.

3 Edge Map Improvement

A method based on local processing is proposed in this paper, which consists of three steps, i.e. edge thinning, edge linking and cracks elimination.

3.1 Edge thinning

After edge detection, most parts of the edge are single-pixel wide. However, because of the noise introduced in the previous image processing stages and the errors introduced in the discrete processing, there are still some parts of edge with more than one pixel in width as shown in Figure 1. In order to convenience the latter processing and improve the computation accuracy, it is very necessary to sharpen the edge lines.

During edge-thinning procedure, the points which cannot be deleted are called *critical points*. The edge point at the centre of a seven-point cluster on mimic Spiral Architecture is the critical point if it meets one of the following three conditions (See Figure 4).

1. The initial linked edge segment will be separated into two or more parts if the central point is deleted;
2. The central point has six neighbouring points marked as edge points;
3. Before edge-linking, the central point is an end point.

An iterative edge-thinning procedure is developed. Moreover, it is a light weight iterative procedure. That means when it finds an edge point which may be deleted according to the criteria, this point is only marked without being deleted immediately. The marked points are deleted after the iteration is completely finished. For example, in Figure 5, if we delete the first detected point as soon as we find that it is not a *critical point* according to the conditions mentioned above, we will lose the second point. Because in Figure 5.b. the second point will be considered as a *non-critical point* if the first detected point is removed. However, using light iterative procedure, we will keep this point as shown in Figure 5.c. Such iterative procedure will avoid producing more breaking points and breaking the initially linked edge lines.

In addition, edge-thinning procedure is executed two times. The first time is before edge-linking (Refer to the next section). The

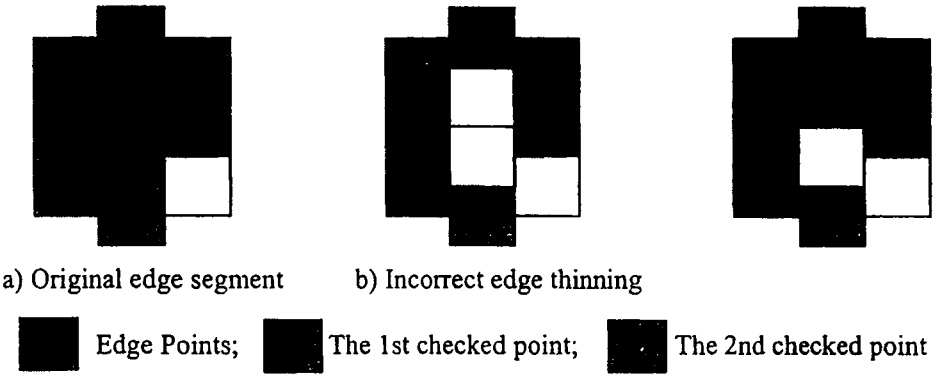


Figure 5 Light weight iterative edge-thinning

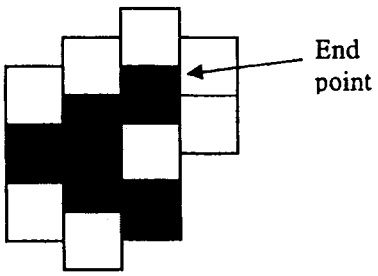


Figure 6 Different treatments to the end points before and after edge-linking

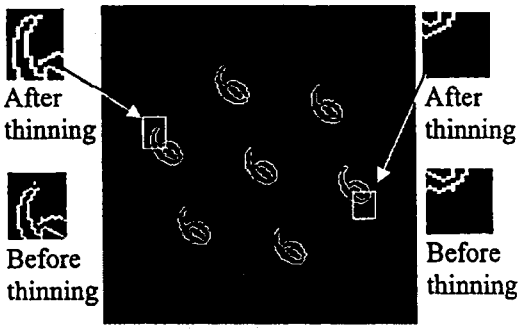


Figure 7 Edge thinning before linking

second time is after edge-linking. The difference is the treatment to the end points (See Figure 6). Before edge-linking the end points are candidate starting points for edge-linking. Edge-thinning will keep them. But they are treated as left error points after edge-linking if they are not linked into any closed edge boundary. Then, they are deleted finally.

Figure 7 shows an experimental result of edge thinning before linking. As shown in Figure 7, unwanted points have been removed to make a thin edge line.

3.2 Edge Linking

If the edges are clear-cut enough, the noise level is low and the edge detection algorithm is supposed to be sufficiently accurate, one can thin the binary edges down to the single-pixel-wide, closed and connected boundaries. Under non-ideal conditions, however, such edges contain gaps that must be filled.

Small gaps can be filled simply by connecting two breaking points directly. In complex scenes with lots of end edge points, however, this method may over-segment the image. To combat the over-segmentation, a heuristic search algorithm is used here. This heuristic search is executed in a 7×7 cluster centred on the reference end-point as shown in Figure 8. This method implies that the possible linking points only appear in the 7×7 cluster centred on the reference end-point. The gaps between two end points out of this range are treated too big to fill in. Moreover, the actual searching area covers only half of the neighbourhood as shown in Figure 8. All the edge point in the searching area including the end points and the points on the edge lines are considered as the candidate points to be tested.

During the heuristic search, a *quality* measure is established. Each pixel in the searching area is assigned a value between zero and one, which stands for the probability of being able to connect to the central point. Five properties of a pixel are considered to measure the point's *quality*. For any candidate point *a* in Figure 8, the quality is a weighed sum of five parts shown

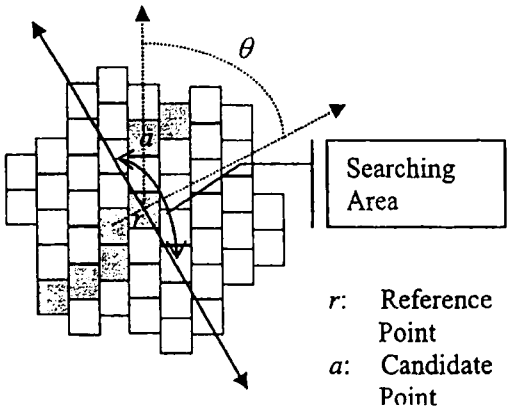


Figure 8 Heuristic searching

as follows.

$$\tilde{q}(a) = w_1\left(1-\frac{dis(a)}{dis_{max}}\right)+w_2\left(1-\frac{ori(a)}{ori_{max}}\right) +w_3\left(1-\frac{gradd(a)}{gradd_{max}}\right)+w_4\left(1-\frac{dird(a)}{dird_{max}}\right) +w_5\left(1-\frac{grayd(a)}{grayd_{max}}\right) \quad (1)$$

where w_i, $1 \le i \le 5$ is weight value, $dis(a)$ is the distance between the reference point and the candidate point, $ori(a)$ is angle θ between the direction of candidate point and the direction opposite to the previous edge point as shown in Figure 8, $gradd(a)$ is the difference of gradient magnitude between the reference point and the candidate point, $dird(a)$ is the difference of the gradient direction between the reference point and the candidate point, $grayd(a)$ is the difference of grey-level between the reference point and the candidate point, dis_{max}, ori_{max}, $gradd_{max}$, $dird_{max}$, $grayd_{max}$ are the maximum of $dis(a)$, $ori(a)$, $gradd(a)$, $dird(a)$, $grayd(a)$. During the procedure, the end edge points in the searching area are given higher priority than the edge points on the edge lines. Namely, it should be more possible to connect an end point to the reference point. Consequently, besides the *quality* value

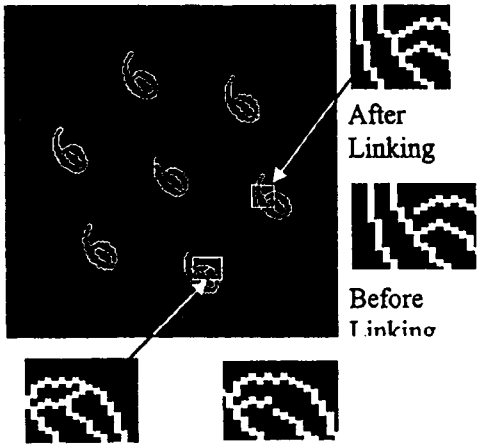


Figure 9 Edge map after the edge-linking

computed according to Equation (1), each end point is given a *bonus* value as Equation (2).

$$q(a) = \begin{cases} \tilde{q}(a) + bonus & a \text{ is an end edge point} \\ \tilde{q}(a) & a \text{ is an edge point on the edge line} \end{cases} \quad (2)$$

Naturally, among all the candidate points, the point which is supposed to connect with the reference point has the highest *quality* value.

The selection of weight values w_i depends on application. It can be estimated through statistical methods. Based on the features of the images in the experiments, we set w_1, w_2, w_3, w_4, w_5 to be 0.25, 0.21, 0.22, 0.22 and 0.1 respectively.

In order to prevent searching correctness from being affected by noise and system errors, a threshold, 0.55, is set for the quality computation before the final decision. Any potential error points, whose quality value is less than 0.55, are ignored.

A shaving operation follows the heuristic search procedure to prune away any remaining end-points, which is edge thinning after linking we mentioned in the previous section. Figure 9 shows the experimental results of edge linking including the second time edge thinning based on the processing results of Figure 7.

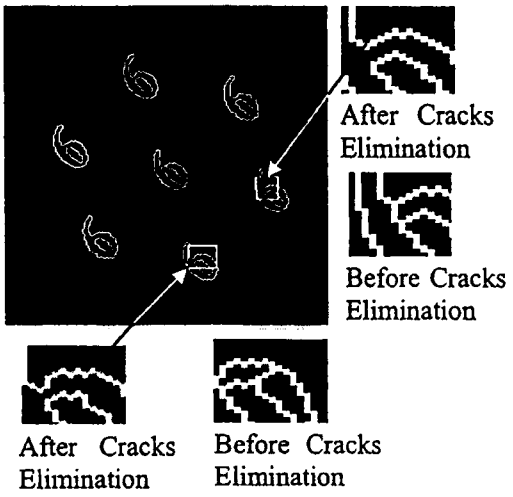


Figure 10 Edge map after cracks elimination

3.3 Cracks Elimination

As shown in Figure 9, the results of edge linking are still not perfect enough. The gaps are really filled, but many cracks are introduced at the same time. Hence, a region, which was a complete one originally, is split to a few parts. This is due to computing errors introduced by Equation (1). To further improve the map quality, and to eliminate the cracks, we must create a technique to merge two separated adjacent parts which have similar properties. In our research, the average grey-level in a region is chosen as the region property to determine whether to merge the adjoining parts together. In the following, we show the procedure for mergence.

Without losing the generality, we can assume that the larger areas normally contain more object features than the smaller areas. Then the checking procedure for the region mergence begins from the largest region to the smallest one, that is, the smaller regions will merge into the adjacent larger one if their properties are similar. The region mergence procedure is shown as follows,

1. Identify all the closed regions, label them and compute their sizes;
2. The largest region is set as the starting reference region for cracks elimination procedure;
3. If the neighbouring smaller region has the similar properties as the larger region, the smaller region merges into the larger one (current reference region). Then, the sizes of the regions will be updated. Go back to step 2;
4. Go to the next larger region and repeat step 3 until all the regions have been checked.

The experimental results after cracks elimination can be seen in Figure 10.

4 Conclusion

In this paper, an edge map improvement method is proposed, which consists of three stages, edge thinning, edge linking and cracks elimination. We inspect five properties of the candidate points in the searching area. We assign a different weight value to each property factor, because the different properties of the point contribute the different influences in determining whether the point will be connected to the current reference end point. Finally, the point to which it is most possible to connect the reference end point will be found.

Compared with Figure 1.b, the quality of edge map on Figure 10 has been improved successfully. Such improved edge map should be beneficial to further image processing.

In the edge map improvement method mentioned above, how to choose the weight values in Equation (1) is an important point which will affect the final edge map quality. In general, it depends on the different images to be processed. We can use the statistical methods to prepare the different sets of weight values for the different catalogues of the images.

In the proposed method, we assume that the candidate points must appear in a 7×7 cluster on Spiral Architecture. To some complicated images, this assumption has to be adjusted accordingly.

In addition, currently we use a straight line to link the reference end point to the candidate point found in the searching area. This will introduce other errors when the images to be processed are more complicated or the gap between two end points is a bit bigger. We are researching other smart methods to improve this part of work.

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

[3] R. M. Haralick, "Digital Step Edges from Zero Crossing of Second Direction Derivatives," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. PAMI-6, Jan 1984.

[4] D. Marr and H. Hildreth, "Theory of Edge Detection," Proceedings of Roy. Soc., London, pp. 187-217, 1980.

[5] V. Torre and T. A. Poggio, "On Edge Detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 147-163, 1986.

[6] D. H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes," *Pattern Recognition*, vol. 13, pp. 111-122, 1981.

[7] Q. Zhu, M. Payne, and V. Riordan, "Edge Linking By a Direction Potential Function (DPF)," *Image and Vision Computing*, vol. 14, pp. 59-70, 1996.

[8] 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.