Proceedings of the 2008 IEEE/ASME
International Conference on Advanced Intelligent Mechatronics
July 2 - 5, 2008, Xi'an, China

EDGE LINKING USING GEODESIC DISTANCE AND NEIGHBORHOOD INFORMATION

Zhijie Wang, Hong Zhang

University of Alberta
Department of Computing Science
Edmonton, Alberta, Canada T6G 2E8

ABSTRACT

This paper deals with the edge linking problem. We propose two improvements to existing algorithms. First we propose the use of an application-specific local neighborhood within which to compute edge direction in order to improve the accuracy of the edge direction. Second, we propose the use of geodesic distance for measuring the proximity between two candidate edge points to be linked, so that the intensity image information can be taken into account. With these two improvements, our edge linking algorithm can make better edge linking decisions than existing algorithms, as demonstrated in the experimental results.

Index Terms— Edge linking, geodesic distance, edge direction

1. INTRODUCTION

An important task in image processing and computer vision is edge processing, which generally consists of two steps, edge detection and edge linking. Edge detection in general is unable to identify all edge pixels that form complete edges, and an edge linking step is therefore needed to create desired complete edges or contours.

There has been much work on edge linking. In [1], Miller, Maeda and Kubo present an algorithm consisting of two steps, labeling and linking, which makes use of a discrete weighting system based upon distance, direction and edge segment's magnitude. Also, Hajjar and Chen present an algorithm [2] based on edge direction and the weak level points. Compatible ending points in the same window with the smallest distance will be linked. Furthermore, Li, Randall and Guan propose to link each ending point to some other point that is considered as its best match according to direction, distance between two points and whether it is an ending point [3]. Additionally, Ghita and Whelan devise an algorithm, referred to as CAEL in this paper, that uses local information around ending points and make linking decisions by minimizing a cost function of direction, distance and whether an edge pixel is an ending point [4].

We observe two areas of improvement in the above edge linking methods. First, when they measure edge direction, only four or eight directions are defined and generally only a small neighborhood of the ending point is used in the analyis. In this paper, we propose to measure edge direction at a finer resolution and show that this can improve the edge linking decision. Second, all the methods use the Euclidean distance to measure promixity between two candidate edge points to be connected, and therefore ignore the intensity information along the connection that should give clue to the plausibility of this connection. In this paper, we employ the geodesic distance in place of the Euclidean distance, in order to make use of the intensity information. With geodesic distance, we expect to improve the edge linking decision.

Related to the edge linking algorithms above, Cook and Delp propose an algorithm [5] in which the edge detection and linking are combined. Therefore, intensity image information is used in edge linking. However, this algorithm models the enhanced image as a random field and the edge path as a random Markov chain. Consequently, it relies on parameters that need to be tuned to obtain desirable results. Finally, edge grouping deals with the problem of connecting the detected edges from an edge detector [6, 7]. However, different from edge linking, edge grouping aims at getting a salient closed boundary of one object, rather than enhancing the edge detection result by connecting short edges into longer edges.

The rest of the paper is organized as follows. In the next section we describe the two key ideas of our edge linking algorithm in detail. We will then present our proposed edge linking algorithm formally in Section 3. Section 4 is dedicated to the experimental evaluation of our algorithm. Conclusion is finally drawn in Section 5.

2. IMPROVEMENTS

In this section, we will talk about two improvements to the existing edge linking algorithms. The first is a new method to measure edge direction. The second is using geodesic distance in edge linking to make use of intensity image information.

978-1-4244-2495-5/08/$25.00 © 2008 IEEE.

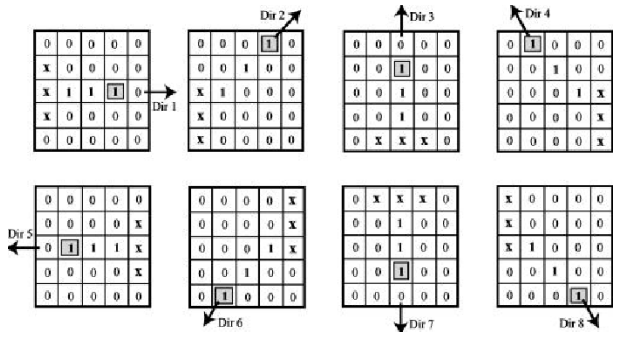


Fig. 1. Edge direction template in CAEL.

Fig. 2. Edge direction calculation in our proposed algorithm.

2.1. Edge direction

In the existing edge linking algorithms, the edge direction is generally measured from a small neighborhood of an ending point. As the example in Figure 1 from CAEL [4], the last three edge pixels of a partial edge are used to obtain the edge direction according to the templates where 0 and 1 represent non-edge pixel and edge pixel, respectively and $\times$ can be either. Unfortunately, eight directions are not sufficient to find the correct result and templates are limited so that they can not cover all the cases. Therefore two kinds of errors are introduced in the direction. The first is from approximating the edge direction by one of those eight directions. The second arises when edge pixels are configured differently from those in the templates.

To address these problems, we propose a new method to measure edge direction. For the ending point P_e, highlighted in Figure 2, we define a window whose size is equal to, g_{max}, the max gap within which we can link an edge. Then we fit a line l to those edge pixels connected to P_e in the window. Between two directions through l, the one closer to $\vec{d_c}$, that is $\vec{d_1}$, is chosen as the edge direction at P_e. $\vec{d_c}$ is the direction from the centoid of the edge pixels to P_e. The centroid is shown as a gray dot in Figure 2. In Figure 2, we also show the edge direction $\vec{d_2}$ measured according to CAEL's method. $\vec{d_1}$ should be more accurate than $\vec{d_2}$. Because g_{max} is set by user with apriori knowledge in different applications, it is a proper window size. As well, our method is more precise as it incorporates all possible directions between 0° and 360° whereas CAEL considers only 8 directions.

2.2. Geodesic distance for edge linking

Euclidean distance is always employed in the existing edge linking algorithms. However, in our proposed algorithm, we use the geodesic distance rather than the Euclidean distance. In the following, we will show the superiority of the geodesic distance to the Euclidean distance.

The geodesic distance between two pixels p and q on an intensity image, $D_g(p,q)$, is defined as follows. Suppose $P = \{p_1, p_2, \cdots, p_n\}$ is a path between p_1 and p_n, where p_i and p_{i+1} are connected neighbors for $i \in \{1, 2, \cdots, n-1\}$. The path length $l(P)$ is defined as

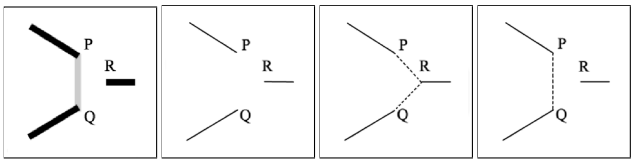
$$l(P) = \sum_{i=1}^{n-1} d_N(p_i, p_{i+1}) \tag{1}$$

i.e., the sum of the pairwise neighbor distances d_N between adjacent points along the path. Then $D_g(p,q)$ is defined as

$$D_g(p,q) = \min\{l(P)\} \tag{2}$$

where P is any path between p and q. A particular geodesic metric is defined by which neighbors are considered to be connected and the values of d_N for each pair of connected neighbors. In our algorithm, pixels are four connected and $d_N(p_i, p_{i+1}) = |I(p_i) - I(p_{i+1})|$, where $I(p_i)$ and $I(p_{i+1})$ are intensity values of p_i and p_{i+1}. Therefore, the geodesic distance considers not only the distance in Euclidean coordinate system but also the intensity difference between the pixels, which obviously contains more information than the Euclidean distance.

Figure 3 shows the difference between using the Euclidean distance and using the geodesic distance. Figure 3(a) is an intensity image with three dark edge segments and a gray segment, where darker pixels corresponding to higher intensity values. Therefore, the gray segment has slightly lower intensity value than the three dark segments. We get the edge image of this intensity image by thresholding followed by thinning. Assume the gray segment does not survive the thresholding processing. Then the edge image will be Figure 3(b). Because the Euclidean distance between points P and Q is farther than that between P and R, then the point P will be linked to R and similarly the point Q will be linked to R as shown in Figure 3(c). However, the geodesic distance between P and Q is shorter than that between P and R. Because there is a path between P and Q with similar intensity value as P and Q. Therefore, we will link P and Q with each other as in Figure 3(d). It can be seen that the geodesic distance can provide more meaningful information about whether two points belong to the same edge than the Euclidean distance.



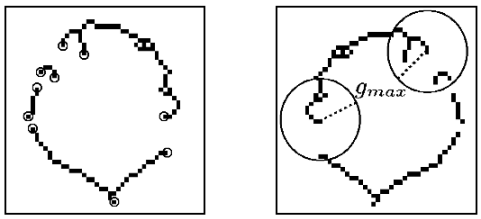
(a) Instensity image (b) Edge image (c) Using Euclidean distance (d) Using geodesic distance

Fig. 3. Differences between using Euclidean distance and geodesic distance

3. EDGE LINKING

We have discussed two improvements to the existing edge linking algorithms in the previous section. Here we will describe our proposed algorithm in detail.

Our edge linking algorithm is extending every edge ending point to the best matching edge pixel. First of all, we find all the ending points in the edge image as in Figure 4(a) where all the terminators are circled. Then we set a maximum gap g_{max}, $i.e.$, for each ending point all the edge pixels within g_{max} will be considered as candidate linking points. In Figure 4(b) we show an example of getting the candidate points for two ending points. In the following step, from each ending point, we pick up the best matching point among all the candidate points and link it to the current ending point.



(a) (b)

Fig. 4. (a) Find ending points, and (b) Find candidate sets of ending points.

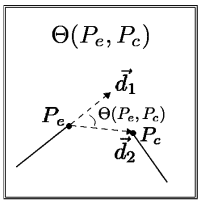


Fig. 5. $\Theta(P_e, P_c)$ term in the likelihood function.

To evaluate candidate points, we propose a likelihood function as follows,

$$H(P_e, P_c) = \frac{1}{D_g(P_e, P_c) \cdot \Theta(P_e, P_c)} \quad (3)$$

```
1   Find all the ending point in the edge image.
2   For each ending point,
3       Find the candidate points set of the current ending point.
4       Calculate H(P_e, P_c) for every candidate point.
5       If there are any ending points in candidate points set,
6           Find the one with the biggest H(P_e, P_c) among
            all the ending points in candidate points set.
7       Else
8           Find the candidate point with the biggest H(P_e, P_c).
9       End If
10      Link the best matching candidate point found above to P_e.
11  End For
```

Fig. 6. Edge linking algorithm.

P_e and P_c are the current ending point and the candidate point respectively. $D_g(P_e, P_c)$ is the geodesic distance between P_e and P_c which has been discussed in detail in the last section. $\Theta(P_e, P_c)$ is a term evaluating how well P_c matches the edge direction at P_e. As shown in Figure 5, at P_e we can measure the edge direction $\vec{d_1}$ using our method discussed in Section 2. This direction shows the trend where P_e will go. $\vec{d_2}$ in Figure 5 is the actual direction if we link P_e to P_c. The angle between these two directions is $\Theta(P_e, P_c)$. The smaller it is, the better two directions match with each other, the more probably we will link P_e to P_c. The final likelihood $H(P_e, P_c)$ is the reverse of the product of $D_g(P_e, P_c)$ and $\Theta(P_e, P_c)$. The bigger $H(P_e, P_c)$ is, the more probably P_c will be chosen as the best matching point of P_e. In our algorithm, ending points have higher priorities than normal edge pixels when considered to be the best matching point. It means that we first try to pick up a candidate ending point with the biggest $H(P_e, P_c)$. In case there is no ending point in the candidate set, we choose a normal edge pixel candidate with the biggest $H(P_e, P_c)$. After we find the best matching candidate point, we link this one to the current ending point. The same procedure described above will be repeated for each ending point in the edge image. The pseudo-code of the algorithm is presented in Figure 6.

4. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of our proposed algorithm. Among all the existing similar edge linking algorithms described in the introduction section, CAEL is the most representative and has good performance. Therefore, here we will compare our proposed algorithm with CAEL. In the following, we will show that both the new direction measurement method and the use of geodesic distance can benefit edge linking.

4.1. Effect of direction measurement

We run our proposed algorithm using Euclidean distance in the likelihood function (3), so that the only difference from the existing algorithm is our new direction measurement method. We compare the result with that from CAEL, shown in Figures 7 and 8.

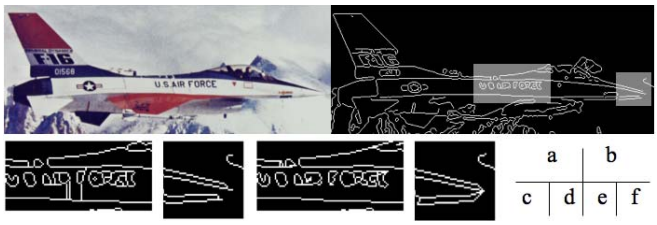


Fig. 7. Comparison between CAEL and our algorithm using Euclidean distance (Plane image).

Figure 7 shows the comparison between the edge linking results of two algorithms on a standard "Airplane" image. Figure 7(a) and (b) are the original intensity image and the corresponding edge image respectively. Owing to limited space, we have omitted the edge linking results of the entire image corresponding to the two algorithms. We have instead included two enlarged parts of the image (highlighted with shaded blocks) illustrating the superiority of our approach. Specifically Figures (c-d) and Figures (e-f) correspond to the results from CAEL and our algorithm, respectively. Comparing Figure 7(c) with Figure 7(e), we can see that CAEL mistakenly links the written characters to the horizontal wing of the airplane. Additionally, comparing Figure 7(d) with Figure 7(f), CAEL can not create the contour of the airplane's nose properly while our algorithm can.

Fig. 8. Comparison between CAEL and our algorithm using Euclidean distance (Pepper image).

Figure 8 is another comparison between the two algorithms on the "Pepper" image. Figure 8(a) and (b) are the original intensity image and edge image, respectively. Again, here we show the examples of differences between the two algorithms, for the highlighted three parts in the edge image. Figures 8(c-h) correspond to edge linking results of CAEL and our algorithm on these three blocks. The pepper stems in Figures (f) and (h) are free of those spurious horizontally connected edges as seen in Figures 8(c) and (e). Additionally, our proposed algorithm can produce the highlighted region on the foreground pepper as an enclosed circile in (g), whereas CAEL can not in Figure 8(d). In view of the above, the superior performance of our proposed algorithm as well as the reason for this superiority are clear.

4.2. Effect of geodesic distance

In the next step, we run our algorithm using geodesic distance instead of Euclidean distance on thea "Airplane" and "Pepper". The corresponding results of five highlighted parts in Figure 7 and 8 are shown in Figure 9. It seems that there is no big difference between using Euclidean distance and using geodesic distance in our algorithm. We attribute this to the fact that our algorithm is able to perform edge linking by carefully computing the edge direction direction. The geodesic distance, although it could be useful, did not bring about additional improvements.

Fig. 9. Result of our algorithm using geodesic distance.

To demonstrate the importance of geodesic distance, we compare (i) CAEL, (ii) our proposed algorithm using Euclidean distance and (iii) our proposed algorithm using geodesic distance, on mineral ore (oil sand) images. Figure 10(a) is an original image, and we want to segment the ore fragment by edge linking. First we threshold the image to get the dark ring around the fragment, and then apply a thinning operation to get the final edge image shown in Figure 10(e). We run the above three different methods (i) to (iii) on this edge image, followed by a post processing step of hole filling. Figure 10(b) and (f) are CAEL's edge linking result and the final segmentation result. Figure 10(c) and (g) are the results of our algorithm using Euclidean distance and Figure 10(d) and (h) are the results of our algorithm using geodesic distance. Here, we can see from Figure 10(b), CAEL produces a much larger object than it should. In Figure 10(c), our algorithm using Euclidean distance produces an improved segmentation. Finally the proposed algorithm using geodesic distance gives us the most satisfactory result. Therefore, using geodesic distance in this example can avoid some errors by making use of intensity image information and benefit the

edge linking.

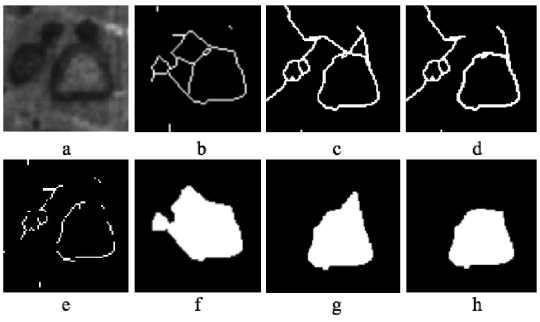


Fig. 10. Comparison of Oil Sand Ore image's segmentation results from CAEL, our algorithm using Euclidean distance and our algorithm using geogesic distance.

5. CONCLUSION

In this paper, we propose a new edge linking algorithm. Two improvements to the existing edge linking algorithms are presented. First, a new measurement method for edge direction can give us more precise and more correct result. Because we use a different window size based on application and we get a continuous direction rather than strict to one of the eight discrete directions. The second improvement results from using geodesic distance between points in image rather than Euclidean distance. Because geodesic distance can provide more meaningful information than Euclidean distance about how likely two points belong to the same edge. Both of these two improvements are helpful to edge linking. Finally, experiments show the superiority of two improvements and the better performance of our proposed algorithm than a representive existing algorithm.

6. REFERENCES

[1] Fredrick L.Miller, Junji Maeda, and Hiroshi Kubo, "Template based method of edge linking using a weighted decision," in *Proceeding of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 1993, vol. 3, pp. 1808–1815.

[2] A.Hajjar and T.Chen, "A vlsi architecture for real-time edge linking," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, pp. 89–94, January 1999.

[3] J.Li, J.Randall, and L.Guan, "Perceptual image processing for digital edge linking," in *Canadian Conference on Electrical and Computer Engineering*. IEEE, 2003, vol. 2, pp. 1215–1218.

[4] O.Ghita and P.Whelan, "Computational approach for edge linking," *Journal of Electronic Imaging*, vol. 11, pp. 479–485, October 2002.

[5] G.W.Cook and Edward J.Delp, "Multiresolution sequential edge linking," in *Proceedings of the International Conference on Image Processing*. IEEE, 1995, vol. 1, pp. 41–44.

[6] Joachim S.Stahl and Song Wang, "Edge grouping combining boundary and region information," *IEEE Transactions on Image Processing*, vol. 10, pp. 2590–2606, October 2007.

[7] A.Blake P.Perez and M.Gangnet, "Jetstream: Probabilistic contour extraction with particles," in *Proceedings of the Eigth IEEE International Conference on Computer Vision*. IEEE, 2001, vol. 2, pp. 524–531.