2010 International Conference on Intelligent Computation Technology and Automation

Image Segmentation Based on Global Extraction and Local Repair of Boundaries

He Mincong, Ling Weixin, Zhao Jianhui

South China University of Technology, Guangzhou, Guangdong, 510640, China
zerohit@163.com

***Abstract*—As the edge lines obtained from the Canny's edge detection operator are not closed, it difficult to use the edge map for the image segmentation. Moreover, the detection result contains some edge lines have nothing to do with the target partition, also resulting in poor segmentation results. This paper presents a method combining global edge extraction with local edge repair, in conjunction with the morphological processing to obtain closed edge lines and remove irrelevant edge. GELR is short for this method.**

Keywords-* *Image segmentation; Edge detection; morphology; Canny operator

I. INTRODUCTION

Image segmentation is to split a physical connectivity object from the original image. There are two main techniques for image segmentation: region-based techniques and edge-based techniques.

Thresholding [1] and region growing [2,3] are two main method of region-based techniques. Thresholding method usually considers the entropy value as a judgment for the selection of threshold value. The dimension of threshold can be one-dimensional or multidimensional, higher dimension thresholds contain more information, and its result is better, but the computation of it will increase accordingly.

Another typical segmentation technique is edge-based methods [4]. The traditional method has the advantages of accuracy positioning and fast operation. However the edge lines obtainned are usually not closed. Another kind of edge detection method is the level set method (LSM), which is present by Osher and Sethian in 1988 [5]. It is based on a geometric contour model. However for larger resolution images, the computational speed of LSM is too low. In order to improve the evolution speed of the level set, a number of improved algorithms [6,7] have been proposed. However, these algorithms are complex, and the computational cost of them is still large. Medical image segmentation is usually resort to employing LSM [8].

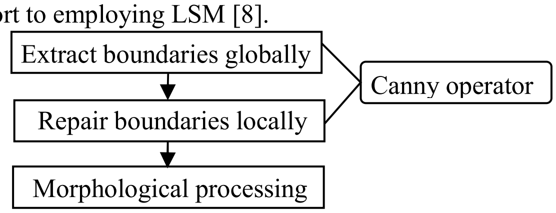


Figure 1. The flowchart of GELR.

The segmentation algorithm described in this paper is an edge-based method. Canny operator [9] is applied in the main processing. The typical shortcoming of edge-based methods is the difficulty in obtaining closed boundaries. In order to approach this problem, we propose to apply Canny operator for global edge extraction and local edge repair. Through these operations the target's edge attain certain density, while the irrelevant edge is attenuated. And then morphological processing is applied to extract the target region and remove the others. Flowchart is shown in Fig.1.

II. CANNY OPERATOR

In the edge detection, differential operators are frequently used for mutations detection in the region of a pixel. However, in the image, the differential of edge and noise is similar, so differential operator are noise sensitive. In order to avoid the effect of noise, a low-pass filter is used to attenuate the high-frequency components in the image signal firstly, and the Gaussian low-pass filter often has excellent performance in practice. Here is a two-dimensional Gaussian function whose covariance is zero:

$$G(x,y)=\frac{1}{2\pi\sigma^2}exp[-\frac{1}{2\sigma^2}(x^2+y^2)]$$

So when a two-dimensional signal convolves with Gaussian low-pass filter, and then carry out a first-order differential, thus we can get the following results:

$$\nabla(f(x,y)*G(x,y))=f(x,y)*\nabla G(x,y)=f(x,y)*(\frac{\partial G}{\partial x}+\frac{\partial G}{\partial y})$$

The partial differential of and can also be considered as the differential in the horizontal and vertical direction, thus we get:

$$\frac{\partial G}{\partial x}=dg(x)g(y),\frac{\partial G}{\partial y}=dg(y)g(x) \tag{1}$$

$$dg(x)=-\sqrt{k}\frac{x}{\sigma^2}exp(-\frac{x^2}{2\sigma^2}),\ g(x)=\sqrt{k}exp(-\frac{x^2}{2\sigma^2}),\ k=\frac{1}{2\pi\sigma^2}$$

From Equation (1), we know that the two-dimensional filter $\partial G/\partial x$ can be expressed as a one-dimensional Gaussian function multiplies with the first-order differential of Gaussian function. Its function map is shown in Fig.2 (a), the function shape seen in horizontal and vertical direction are shown in Fig.2 (b) (c). It is to say that a two-dimensional signal *f(x,y)* convolving with $\partial G/\partial x$ is equivalent to convolving with *dg(x)* in the horizontal direction and convolving with *g(y)* in the vertical direction. From the view of wavelet decomposition, high-pass filtering is applied in the horizontal direction while low-pass filter is used in the vertical direction. The scaling function of the wavelet decomposition is *g(x)* while the detail function is *dg(x)*, the functions are shown in Fig.3.

After these operations we can get the gradient amplitude. With it, a threshold can be set to eliminate the false edge

978-0-7695-4077-1/10 $26.00 © 2010 IEEE
DOI 10.1109/ICICTA.2010.507

points. And then thin the edge through the two-dimensional interpolation processing.

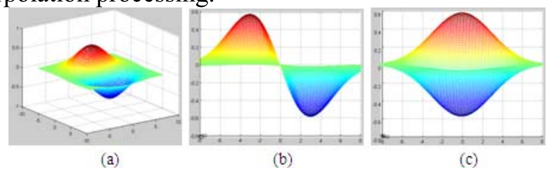


Figure 2. (a) Two-dimensional Gaussian function, (b) the function shape seen in horizontal direction, (c) the function shape seen in vertical direction.

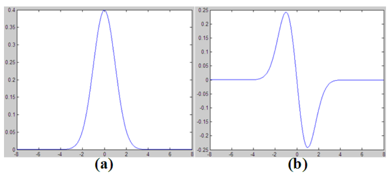


Figure 3. The wavelet decomposition corresponding to the Canny operator: (a) the scaling function, (b) the detail function.

In the image analyzed in this paper, most of the gradient of the edge between target and background is greater than others, so we can adjust the threshold to remove part of the background. However, the threshold selection is difficult. Moreover the boundaries obtained are not closed, even using morphological processing is difficult to make it into a whole.

The approach presented in this paper advocated several processing steps as follows: Firstly remove most of the edge while keeping some key boundaries, which can be attained by Canny operator with an appropriate threshold. And then repair the large gaps in local region. Finally adopt morphological processing to fill the segmentation target while removing most of the background.

III. EXTRACT THE IMAGE BOUNDARIES

The goal of the first step is to obtain most of main boundaries of the target, and remove most irrelevant edge, maybe including some weak main boundaries. Although the Canny operator with larger variance can effectively remove noise or irrelevant edges, it also makes the boundaries deformational, so a small variance is adopt here.

To the traditional Canny operator, the same variances are usually used in the first-order derivative Gaussian function for the horizontal and vertical edge extraction, so the edge information obtained or removed in the two directions is the same. However, in practice, there are some differences in the two directions. So here, the algorithm uses different variances in horizontal and vertical operation, so that most edge of the background will be removed. The following experiments will clarify how this goal will be achieved.

Figure 4. A gray image composed of a group of Gaussian functions, whose variances increase from top to bottom.

Fig.4 is a *512×512* gray image composed of a group of Gaussian functions, whose variances increase from top to bottom. Now use two Canny operators with a variance of 1 and 1.5 respectively to calculate the gradient of each pixel. The 3-D results are shown in Fig.5. As shown in Fig.5 (a) (b), the gradient value calculated by the smaller variance operator is relatively smaller. In the same image, the gradient value is larger where the variance of the Gaussian function is smaller.

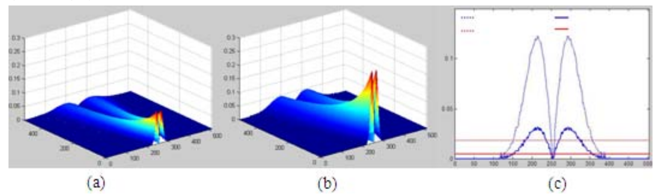


Figure 5. The gradient of Fig.4:(a) the gradient map calculated by Canny operator with variance of 1, (b) the gradient map calculated by Canny operator with variance of 1.5, (c) the cross-section diagram on row 256th of (a)(b) and the threshold lines of 0.15.

And then we get the cross-section diagram on row 256th of Fig.5 (a) (b). The cross-section diagram is shown in (c): solid line corresponding to the operator with variance of 1, dashed line corresponding to the operator with variance of 1.5. Obviously, solid line contains more small fluctuation, indicating that operator with small variance has a better extraction of the details. If adopt 0.15 as the threshold, there are two different thresholds, such as the straight line shown in the Fig.5 (c). A larger variance is corresponding to a larger threshold value. Through this experiment we can speculate that if utilizing different variances, the horizontal variance and the vertical variance, at the same time, the operator can remove most edge in a direction (the direction of the smaller variance) utilizing a suitable threshold. The next experiment will illustrate this in detail.

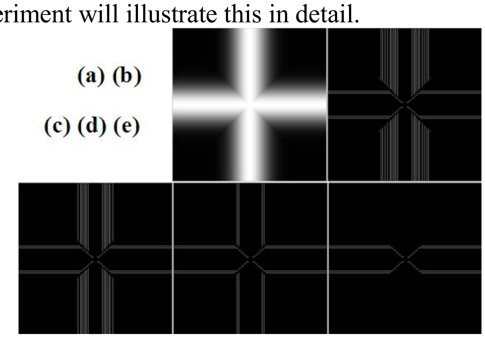
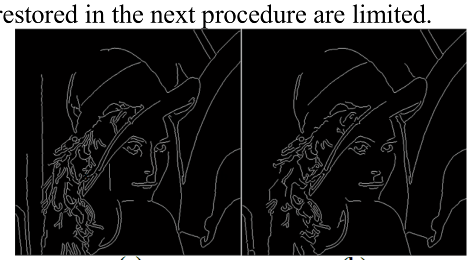


Figure 6. Edge maps from different parameter: (a) two orthogonal beams superimposed by two Gaussian functions with a variance of 4. (b) (c) (d) (e) use the Canny operator with a horizontal variance of 1 and vertical variance of 2 to extract the edge from (a), and the thresholds of them are [0.08 0.1], [0.1 0.2], [0.1 0.25], [0.2 0.3] respectively.

Fig.6 (a) is a gray image generated by two Gaussian functions. Now extract the edge by Canny operator. And Fig.6 (b) (c) (d) (e) show the results adopting different thresholds. In Fig.6 (b) (c) (d), obviously, there is more detail in the vertical direction (corresponding to the horizontal variance). And as the threshold increases, more vertical edges have been removed, while the main vertical edges and most horizontal edges are preserved. When the threshold becomes larger as in Fig.6 (e), the vertical edges are removed completely, keeping the horizontal edges. Experiments show that using different variances in the two directions can effectively remove the edge lines in a direction (corresponding to a smaller variance).

Take this conclusion to process a *512×512* Lena gray image. The results are shown in Fig.7. By comparison, more vertical edges in (b) have been removed, while the horizontal edges have been preserved better than (a). For instance, at the head of the cap in (b), the edges have a greater extension, which is more conducive for the next step. And most of the background edges are removed in (b), so the edges which can be restored in the next procedure are limited.



(a) **(b)**

Figure 7. Edge maps of LENA image: (a) edge map extracted by Canny operator with a variance of 1 and threshold interval of [0.1 0.3], (b) edge map extracted by Canny operator with a horizontal variance of 1, a vertical variance of 1.5 and a threshold interval of [0.1 0.3].

IV. BOUNDARIES REPAIR

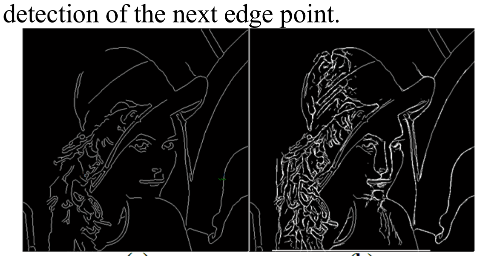
The goal of this step, basing on the edge map of the previous step, is to repair the edges locally so that the edge lines around the boundaries are intensive enough for the next processing. Repair operation is composed by two steps. Firstly, detect discontinuous regions of the edge map row by row, whenever the detected region is in need of restoration, then to repair it. The specific operations are as follows:

TABLE I. THE PSEUDO-CODE OF THE BOUNDARIES REPAIR

```
For each row j in BW
 (BW is the edge map of the previous step)
  the left part:
    select a queue q for edge points
      with an interval in left part of row j
    For each point q[i] in q
    detect the M×N domain of q[i] in BW
    if the M×N domain is need repair
      repair_region=L×W domain of q[i] in original image
      R=CannyOperator(repair_region,variances,threshold)
      update BW with R
  the right part:
  (the same procedure with different variances and threshold)
```

Step 1: Search for edge points of the image row by row from top to bottom, and when an edge point is found, consider its *M×N* domain as the detection region. In order to reduce duplication of detection and computational cost, a searching interval will be added between the edge points. For instance, when an edge point is found, the next one that would be added into the detecting queue must be *N/2* away from it (this value could be set appropriately). Because the procedure is from top to bottom, detect the bottom half of the *M×N* region whether there are a sufficient number of edge points. If the number of edge points is less than a threshold value, go to the next step to repair it. Otherwise, search for the next edge point.

Step 2: If the detection region of *(i,j)* need repair, select the *L×W* domain of *(i,j)* in the original image as the repair region (For instance, the *32×32* domain). In this region, use Canny operator to extract the edge, and the result will be immediately updated back to the *512×512* edge map so as to reduce the detection times. After the processing, go to step 1 for the detection of the next edge point.



(a) **(b)**

Figure 8. Edge maps: (a) edge map obtained in the previous step, (b) edge map after repair: a Canny operator with a variance of 4 and a threshold interval of [0.1 0.2] will be adopt for the left part, while the other uses the operator with a variance of 5 and a threshold interval of [0.2 0.4].

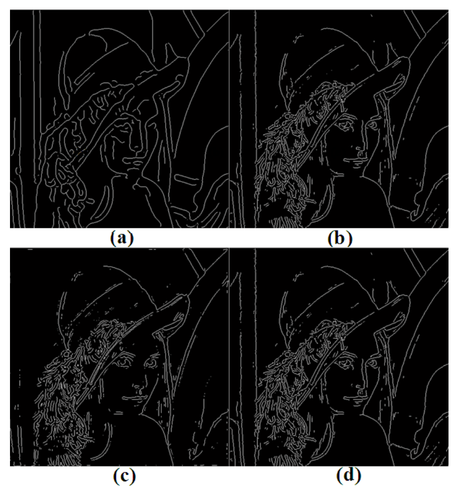


Figure 9. Edge maps from different operators: (a) edge map extracted by Canny operator with a variance of 3 and a threshold of 0.1, (b) edge map extracted by Prewitt operator with a threshold of 0.03, (c) edge map extracted by Log operator with a variance of 1.5 and a threshold of 0.006, (d) edge map extracted by Sobel operator with threshold of 0.03.

The pseudo-code of this processing is shown in table 1. In some images, the left part is quite different from and the right part, so the two parts of the image will be processed separately, only utilizing different parameters in Canny operator, but the same steps. On the other hand, because the edge of the target's boundaries is more clearly than the others, a operator with a relatively larger variance would be applied for the repair. In this procedure, some of the edge lines of the background may get restored, but the operation of the previous step has removed most of them, making the restored edges unable to form a closed region. Hence most of these edges will be cleaned when go through the morphological processing.

For the processing of Lena image, the repair result is shown in Fig.8 (b). Obviously the edge lines are dense enough around the boundaries of the segmentation target. At the same time an edge line would be added at the bottom of the image before the next processing to avoid the discontinuousness of the boundaries, for the location of the target alongside the image border (bottom). If the location of the target is not alongside the image border, the edge line will be attenuated and removed by the morphological processing.

In Fig.9, several operators are used for the edge extraction of Lena image. Compared with Fig.8 (b), the edges extracted by these operators contain more noise, and some of boundaries have a big gap. In addition, the edge features of the target are more clearly in Fig.8 (b).

V. IMAGE SEGMENTATION

In the image obtained from the previous step, the gap of the target's boundaries is small enough that it can be closed by the morphological processing. The following processing uses the Matlab functions: bwmorph and bwfill which is a seed-fill function. The operations are as follows (the word in the bracket is the command of Matlab): twice dilation (dilate) -> once bridge-join (bridge)-> twice shrink operation (shrink) -> once dilation->holes filling (holes)-> three times erosion (erode)-> once dilation. Procedure is shown in Fig.10.

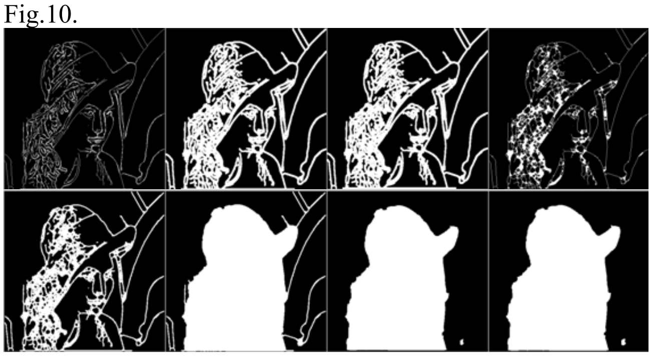


Figure 10. the edge map obtained in the previous step and the result of the 7 morphological steps.

Figure 11. the original image and the segmentation result.

In this procedure, the small gap of boundaries can be bridged by some thin lines after the twice dilation and once bridge-join operation. And the thin lines can't be removed or cut by the following twice shrink operation. However, for the large gap, after these operations is still disconnected, and will be removed by the erosion, that is why the edges of the background can be cleaned.

Finally, multiplying the original image with the binary image obtained from this processing, we get the segmentation object, shown in Fig.11.

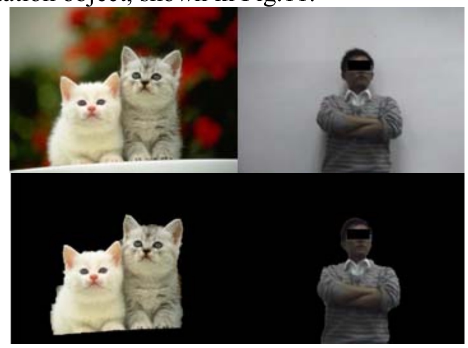


Figure 12. The original image and the segmentation results.

VI. CONCLUSIONS

This paper presents an image segmentation method based on edge detection, and describes its three main steps in detail, including the global boundaries detection, local repair and the morphological processing. In the experiment the algorithm can split the target from background, with less noise. The future work is to study how to enhance the robustness of the algorithm, as well as adjust parameters by better strategy.

ACKNOWLEDGMENT

This work was supported by a grant from the National High Technology Research and Development Program of China (863 Program) (No.2007AA02Z482), by a grant from the Major Programs of Guangzhou (No. 2007C13G0131).

REFERENCES

[1] Orlando J. Tobias, Rui Seara, Image Segmentation by Histogram Thresholding Using Fuzzy Sets, IEEE Transactions on Image Processing, Vol.11, No.12, December 2002, pp. 1457-1465.

[2] Adams R., Bischof L. Seeded region growing. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1994, 16 (6), 641–647.

[3] Tremeau A, Borel N. A region growing and merging algorithm to color segmentation. Pattern Recognition, 1997, 30(7):1191-1203.

[4] Barkhoda W., Tab F.A., Shahryari O.-K., Fuzzy edge detection based on pixel's gradient and standard deviation values. Computer Science and Information Technology, 2009. IMCSIT '09. International Multiconference on 12-14 Oct. 2009, Page(s):7 – 10

[5] Osher S, Sethian J A. Fronts propagating with curvature dependent speed: algorithms based on Hamilton2Jacobi formulations[J]. Journal of Computational Physics, 1988, 79 (1) : 12～49.

[6] Sethian J A. Curvature and evolution of fronts [J]. Commun. Math. Phys. 1985, 101: 487-499

[7] Sethian J A. A fast marching level set method for method for monotonically advancing fronts [C]. In Proc. Nat. Ac Science, 1996: 1591-1694.

[8] A.Yezzi, S.Kichenassagmy, A.Kumar, P.Olver, and A.Tannenbaum. A geometric snake model for segmentation of medical imagery. IEEE Trans on Medical Imaging, 1997,16(2): 199-209.

[9] J. Canny. "A computational approach to edge detection," IEEE Trans. on Pattern Analysis and Machine Intelligent, Vol. 8, pp. 679-698, 1986.