Line Extraction Using Image Carving

Zhanggui Zeng and Hong Yan
School of Electrical and Information Engineering,
University of Sydney, NSW 2006 Australia
zzeng@ee.usyd.edu.au

Abstract

A line extraction method based on image carving is presented in this paper. The gradient magnitude of a pixel is taken as the point-importance of the pixel. The length of the line where the pixel lies, is considered as the line-importance of the pixel. Iteratively, those pixels, whose point-importance value is less than an updated point-importance threshold and whose line-importance value is less than an updated line-importance threshold, are carved off. In other cases, the pixels are preserved. A broken line can be identified and repaired using the gradient image of the last iteration. This method can extract "important" lines, remove "unimportant" ones, and repair broken ones.

Keywords: Image segmentation; Edge filter; Line extraction; Image carving.

1 Introduction

Line extraction is very important to object recognition and tracking. There are many effective line extraction methods (Gates, Haseyama, and Kitajima, 1999, Pires, De Smet, and Bruyland 2000). They may be classified into two classes: line extraction from edges, and line extraction from regions. The lines extracted from the edges can be taken as edge lines, while the lines extracted from the regions can be taken as region boundary lines or contours for gray images, or skeletons for binary images. The Hough transform is a well-known line extraction method. It often runs on binary edge images in order to reduce its expensive computation (Arcelli and Ramella 1995). This paper proposes a line extraction method based on the two importance values of each pixel, i.e. point-importance values or gradient magnitudes and the line-importance values detected by a line searching method. Iteratively, these pixels, whose point-importance magnitudes and line-importance values are less than two updated thresholds, are carved off. Hence the important lines are preserved. This method can extract the long and light lines, keep the connectivity of important lines, and repair broken lines by comparing the two gradient images of two adjacent iterations.

2 Initial Carving

Image carving is based on a hypothesis that an "important" line is long and built by continuous important pixels, and an "unimportant' line is short and formed by unimportant pixels. Hence the proposed carving is expected to pick up the "important" lines, and clean away the "unimportant" lines.

The aim of image carving is to carve away the unimportant pixels and preserve the important pixels. The point-importance of a pixel can be represented by its maximal directional gradient to 8 neighbors, i.e.

$$I_p(x,y) = Max\{0,\ f(x,y) - f(x+u_i, y+v_i) \mid i = 0,1,...,7\}$$

where $f(x,y)$ represents the pixel intensity at coordinate (x,y); $f(x+u_i, y+v_i)$, $i = 0,1,...,7$ represent the 8 neighbors around the pixel $f(x,y)$; i denotes a direction index and zero is taken to prevent negative importance value. In some cases, two adjacent pixels own the same point-importance value. To distinguish their importance, we define a high order point-importance, i.e.

$$I_p^{k+1}(x,y) = Max\{0,\ I_p^k(x,y) - I_p^k(x+u_i, y+v_i) \mid i = 0,1,...,7\},$$
$$k = 1,2,...$$

where $I_p^k(x,y)$ denote the kth order point-importance of pixel $f(x,y)$, and $I_p^k(x+u_i, y+v_i)$, $i = 0,1,...,7$ represents the 8 neighbors of $I_p^k(x,y)$. When the two pixels achieve different point-importance at the least order k, the one that has greater importance value may be taken as more important than the other one. The line-importance of a pixel $I_p(x,y)$ can be measured by the length of the line where the pixel lies, i.e. $I_l(x,y) = Len(x,y)$ where Len is a hypothetic detector for the length of the line going by (x,y).

We first carve away the neighboring pixel whose point-importance is the least among the non-zero neighbors around the pixel $I_p(x,y)$, i.e.

$$I_{p\min}(x+u_{i*}, y+v_{i*}) = Min\{I_p(x+u_i, y+v_i) \mid I_p(x+u_i, y+v_i) > 0, i = 0,..7\},$$

$$C_0(x,y) = Zero\{I_p(x+u_i, y+v_i) \mid i = 0,...,7\},$$

$$I_p(x+u_{i*}, y+v_{i*}) = 0 \text{ if } I_p(x,y) > I_p(x+u_{i*}, y+v_{i*})$$
and $C_0(x,y) > 4$

where direction index $i*$ points to the least important neighbor $I_{p\min}(x+u_{i*}, y+v_{i*})$ from the pixel $I_p(x,y)$, and $Zero$ is a counter to zero neighboring pixels around the pixel $I_p(x,y)$. The use of counter

$Zero$ is to prevent some important lines being broken. The carving at $I_p(x, y)$ should be prohibited when the number $C_0(x, y)$ of non-zero neighbours of the pixel is less than 4. The carving operation based on Equation 5 is carried out on every pixel of the image $I_p(x, y)$ in a natural order.

The above initial carving is not uniformly processed over the image. So we thin all lines to single-pixel lines by removing idle points. The idle pixel of a line can be detected by testing its zero region number and zero number, i.e.

$$C_r(x, y) = Region\{I_p(x + u_i, y + v_i) \mid i = 0,...,7\},$$

$$C_0(x, y) = Zero\{I_p(x + u_i, y + v_i) \mid i = 0,...,7\},$$

$$I_p(x, y) = 0 \text{, if } C_l(x, y) = 1 \text{ and } C_0(x, y) \neq 7$$

where $Region$ is a counter to zero regions separated by non-zero pixels, $Zero$ is a counter to zero neighbouring pixels around the pixel $I_p(x, y)$. Figure 1 illustrates some examples of $Region$ counter and $Zero$ counter. If $C_r(x, y) = 1$ and $C_0(x, y) \neq 7$, then the pixel is an idle pixel of a line and can be removed. Some possible thinning examples are shown in Figure 2. The line thinning based on Equation 6 is also carried out on every pixel of the image $I_p(x, y)$ in a natural order.

3 Carving at one iteration

After the initial carving on the gradient image $I_p(x, y)$, some lines may be broken, i.e. some weak linking pixels on lines are carved off incorrectly. To fix a broken line, we have to find the two endpoints of the broken line. The two endpoints satisfy requirements of $C_r(x, y) = 1$ and $C_0(x, y) = 7$. At the endpoints of the broken line, we search for the missing points by comparing the previous gradient image $I'_p(x, y)$ of the iteration $t - 1$ with the current gradient image $I_p(x, y)$. For the initial carving, i.e. $t = 0$, the image $I'_p(x, y)$ refers to the previous gradient image $I_p(x, y)$ to the initial carving. From one endpoint of the line, we apply a forward searching on the previous gradient image $I'_p(x, y)$ for a missing pixel (as illustrated in Figure 3). The direction index $d*$ of the forward searching points to the endpoint from the pixel next to the endpoint. The forward searching ranges from $(d*-2)\ MOD\ 8$ to $(d*+2)\ MOD\ 8$. MOD function is used for confining the direction index between 0 and 7. A missing point satisfies $I'_p(x + u_j, y + v_j) \neq 0$ where j is a direction index between $(d*-2)\, MOD\ 8$ and $(d*+2)\, MOD\ 8$. To repair the broken line, the missing point can be added to the current gradient image, i.e.

$$I_p(x + u_j, y + v_j) = I'_p(x + u_j, y + v_j).$$

The restored missing point now becomes an endpoint of the broken line. Thus, the broken line can be repaired by repeating the above action. However, the repairing should be stopped when the missing point is a crossing point in the previous gradient image $I'_p(x, y)$ or the missing point meets a point of another line in the current gradient image $I_p(x, y)$. Otherwise, the repairing will restore all the pixels which are carved off in the current iteration.

After the initial carving, including line thinning and broken line repairing, we turn the gradient image to the one-pixel-thick line image.

In one iteration t, we carve away those pixels whose point-importance values are less than, or equal to, the point-importance threshold $T_p(t)$ and whose line-importance values are less than, or equal to, the line-importance threshold $T_l(t)$. To check the line-importance of a pixel may cost much time when the pixel lies on a long line. In practice, for those pixels whose point-importance magnitudes are less than, or equal to, the point-importance threshold $T_p(t)$, we can check if their line-importance magnitudes are greater than the current line-importance threshold $T_l(t)$.

Lines may be classified into two classes, isolated lines and edge lines. There is no cross point or branch line in the isolated line within the length $T_l(t)$. One or more cross points and branch lines are on the edge line within the length $T_l(t)$. However, all branch lines are on one side of the edge line. The method of testing the line-importance of pixel $I_p(x, y)$ on an isolated line is different from the method of testing the line-importance of pixel $I_p(x, y)$ on an edge line.

After some pixels are carved off, some lines may not be one-pixel-thick anymore. Therefore, those lines have to be thinned. In addition, some important points may be removed incorrectly and some lines may be broken. Hence, the line repairing should be reapplied.

4 Iterative carving

In iterative carving, the two thresholds $T_p(t)$ and $T_l(t)$ should be updated with iteration t. Otherwise the carving will fail. We define a variable carving step size as $\Delta T_p(t)$, given as

$$\sum_{k=T_p(t)}^{T_p(t)+\Delta T_p(t)} H(k) \approx \gamma \sum_{k=I_{p\min}}^{I_{p\max}} H(k), \quad \Delta T_p(t) \geq 1$$

where $H(k)$ is the histogram function of the gradient image $I_p(x, y)$, γ is a coefficient between 10^{-3} and 10^{-1}, $I_{p\max}$ is the maximum of the gradient image and $I_{p\min}$ is the minimum of the gradient image. Some examples of carving step sizes are shown in Figure 4. The point important threshold can be updated as

$T_p(t+1) = T_p(t) + \Delta T_p(t)$. When $t = 0$, $T_p(0) = I_{p\min}$. When $T_p(t) > I_{p\max}$, stop the carving.

It is reasonable to expect a preserved pixel with a small point-importance to have a large line-importance, and a preserved pixel with a large point-importance to have a small line-importance. Furthermore, we suggest that the line-importance threshold be updated as an exponential function of point-importance, i.e.

$$T_l(t) = \beta \log_2 W \log_2 H \, Exp(-\frac{(T_p(t) - I_{p\min})^2}{\delta^2}).$$

In the process of iterative carving, First, we turn a gray image $f(x, y)$ into a gradient image $I_p(x, y)$. Then, based on each gradient pixel, the least important neighbouring pixel is carved off. As a result of line thinning, the gradient image becomes one-pixel-thick line networks. Line repairing restores broken lines. During the iterative carving, the point-importance threshold and line-importance threshold are updated. Those pixels, whose point-importance values are less than the point-importance threshold and whose line-importance values are less than the line-importance threshold, are carved off. The remaining pixels are preserved at the current iteration. The line-importance of a pixel is tested by searching for the length of the line where the pixel is embedded. At the end of each iteration, line thinning and line repairing are applied in order to restore the broken lines.

5 Experiments and discussion

The proposed line extraction method is applied to several test images (as illustrated in Figure 5 and Figure 6). To verify the validation of the proposed method, a thinning algorithm is applied to the same gradient images following thresholding. The thinning yields one-pixel-thick lines from the thresholding gradient images. In comparison with the thinning algorithm, image carving has two principle advantages. It is able not only to preserve the long and light lines, but also to keep the connectivity of the lines. The thinning method preserves unimportant lines when the threshold is low. When the threshold is high, it yields broken lines. The carving computation time for a 512x512 image takes about 15 seconds using a 355 MHz Pentium II computer.

This algorithm involves three parameters, α, β and γ. They can be adjusted in terms of users' requirements. However, the method has a drawback when the line-importance threshold is an exponential function of point-importance threshold. It cannot extract a slope edge line successfully. When several adjacent pixels on the slope edge have high gradient magnitudes, these pixels cannot be removed because low line-importance is allowed for pixels with high point-importance. More line-importance threshold functions or adaptive line-importance threshold should be investigated in our future research. Furthermore, the line extraction algorithm can be speeded up by recording and making use of searched pixels.

6 References

Arcelli, C. and Ramella, G. (1995): Finding gray-skeleton by iterated pixel removal, *Image and Vision Computing*, **13**(3):159-167.

Gates, J. Haseyama, M. and Kitajima,H. (1999): A real-time line extraction algorithm, *Circuits and Systems*, ISCAS'99, Proceedings of the 1999 IEEE Int. Symposium on, **4**:68-71.

Pires, R. L. De Smet, P. and Bruyland, I. (2000): Line extraction with the use of an automatic gradient threshold technique and Hough transform, *Image Processing*, Int. Conf. proceedings, **3**:909-912.

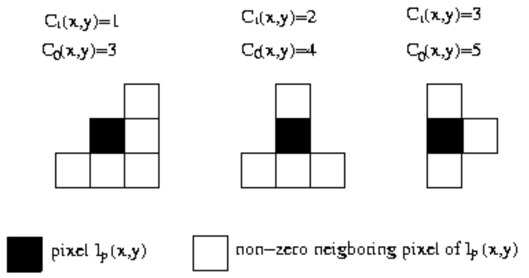


Figure 1. Examples of Region counter and Zero counter.

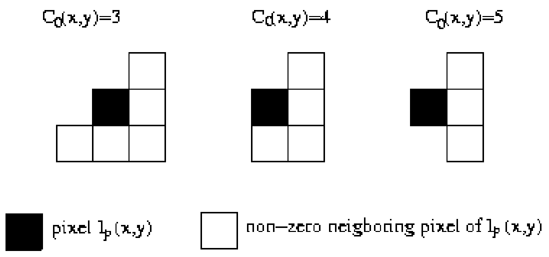
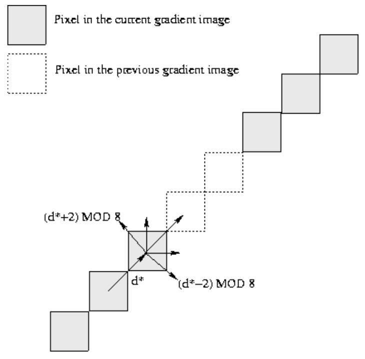


Figure 2. Possible line thinning examples.



Pixel in the current gradient image
Pixel in the previous gradient image
(d*+2) MOD 8
d*
(d*–2) MOD 8

Figure 3. A forward searching for a missing pixel.

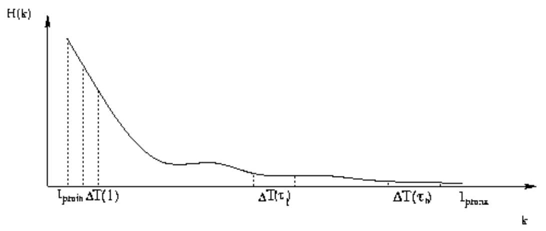
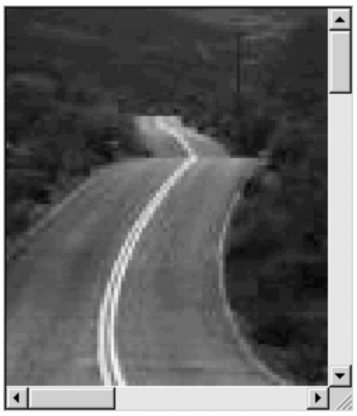
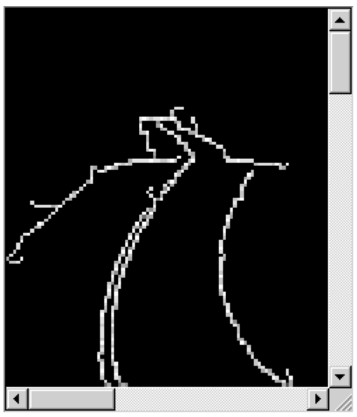


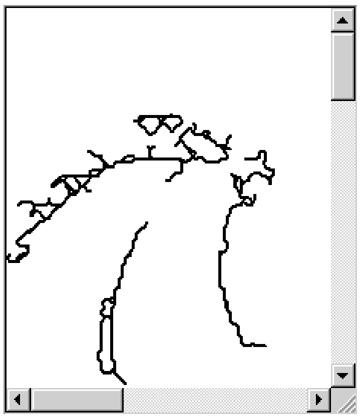
Figure 4. The update of carving step size.



(a)

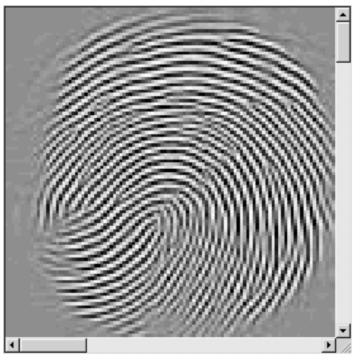


(b)

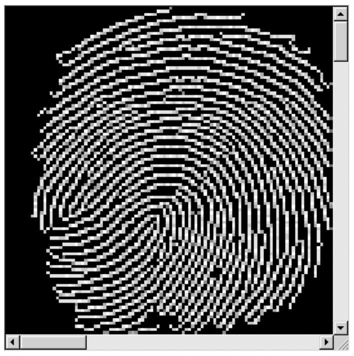


(c)

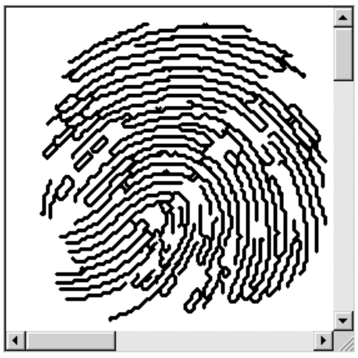
Figure 5. The carving results and thinning results of a "road", (a) original image, (b) carving results, α =0.15, β =2.5, γ =0.01, (c) thinning results, gradient threshold 30.



(a)



(b)



(c)

Figure 6. The carving results and thinning results of a "fingerprint", (a) original image, (b) carving results, α =0.3, β =0.8, γ =0.01, (c) thinning results, gradient threshold 20.