ME 5405
Machine Vision

Image Segmentation

KB Lim
Mechanical Engineering Department
National University of Singapore
mpelimkb@nus.edu.sg

Segmentation - Introduction

1. <u>Introduction</u>

Segmentation is generally the first step in image analysis. **<u>It sub-divides an image into its constituent parts or objects</u>**. The sub-division should stop when the objects of in interest in an application have been isolated.

Example:

In automatic air-to-ground target acquisition applications, we are only interested, amongst other things, vehicles on a road.

We must first segment the captured image to isolate the road, and then segment the road down to objects of a range of sizes correspond to potential vehicles. We should not segment below this scale, nor anything outside the road.

Segmentation algorithms for monochrome images generally are based on one of the two basic properties of gray-level values:

Discontinuity and Similarity.

Segmentation - Introduction

1. Introduction

Discontinuity

The approach is to partition an image based on abrupt changes in gray-level. The principal areas of interest within this category are the detection of isolated points and detection of lines and edges in an image.

Similarity

This approach is based on thresholding, region growing, and region merging and splitting.

Segmentation

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Fig. 1. A general 3 x 3 mask.

2. Detection of Discontinuities

Techniques for the detection of three basic types of discontinuities in a digital image: points, lines and edges are presented below:

The common way is to run a mask shown in Fig 1 through the image. If z_i's denote the gray-level values of the pixels under the mask , the response of the mask at any point in the image is:

$$R = w_1 z_1 + w_2 z_2 + \ldots\ldots\ldots\ldots + w_9 z_9$$

$$R = \sum_{i=1}^{9} w_i z_i \qquad (1)$$

R is defined with respect to the central position of the mask, and *R* is computed for boundary pixels with the use of appropriate partial neighbours.

Segmentation

2.1 Point Detection

-1	-1	-1
-1	8	-1
-1	-1	-1

Fig. 2. A mask used for detecting isolated points different from a constant background.

Using the mask shown in Fig 2, we can say that an isolated point at the location on which the mask is centered if

$$|R| > T \qquad (2)$$

where *T* is a non-negative threshold, and *R* is given by Eq. (1)

With the coefficients in the mask, the gray level of an isolated point will be quite different from its neighbours after computation.

Note that the same mask is used for spatial filtering, but the emphasis here is that only the differences which are large enough (> *T*) are considered as isolated points.

Segmentation

2.2 Line Detection

(a) Horizontal

-1	-1	-1
2	2	2
-1	-1	-1

(b) +45°

-1	-1	2
-1	2	-1
2	-1	-1

(c) Vertical

-1	2	-1
-1	2	-1
-1	2	-1

(d) -45°

2	-1	-1
-1	2	-1
-1	-1	2

Fig. 3 Line mask

If mask (a) in Fig 3 is moved around an image, it would respond more strongly to line (one pixel thick) oriented horizontally. With constant background, the maximum response would result when the line passes through the middle row of the mark.

1	1	1	1	1	
5	5	5	5	5	R in this row would be high
1	1	1	1	1	

Segmentation

2.2 Line Detection

Mask:
- (b) responds best to lines oriented at 45^0.
- (c) for vertical lines
- (d) for lines oriented at -45^0.

Note: that in these masks, coefficients in the preferred directions are given higher weightage.

Let *Ra, Rb, Rc* and *Rd* denote the responses of the masks in Fig 3, from left to right, where the *R*'s are given by Eq. (1).

Suppose that all masks are run through an image. If, at a certain point in the image. $|R_i| > |R_j|$, for all $i \neq j$, that point is said to be more likely to associate with a line in the direction of mask *i*.

Example, If $|Rb| > |Rj|$ for $j = a, c, d.$

that particular point is said to be more likely associated with a 45^0 line

Segmentation

2.3 Edge Detection

An edge is the boundary between two regions with relatively distinct gray-level properties.

Assumption: The regions in question are sufficiently homogeneous so that the transition between two regions can be determined on the basis of gray-level discontinuities alone.

The idea underlying most edge-detection techniques is the computation of a local derivative operator.

Fig 4(a) shows a light strip on dark background

- Profile of gray-level at the edge is modelled as a smooth change of gray-level.
- First derivative of the gray level profile is positive at the leading edge of a transition, negative at the edge, and zero in areas of constant gray level.
- The second derivative is positive for that part of the transition associated with the dark side of the edge, and negative for that part of the transition associated with the light side of the edge.

Segmentation

2.3 Edge Detection

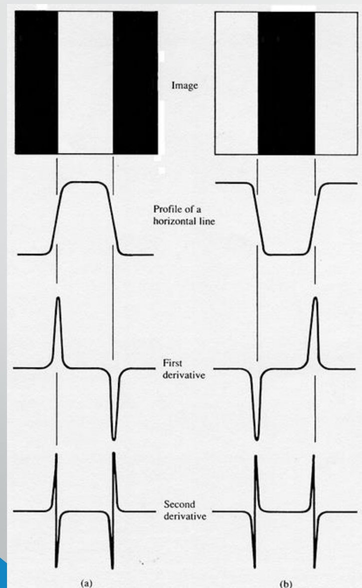


Fig 4(a) shows a light strip on dark background

- Profile of gray-level at the edge is modelled as a smooth change of gray-level.
- First derivative of the gray level profile is positive at the leading edge of a transition, negative at the edge, and zero in areas of constant gray level.
- The second derivative is positive for that part of the transition associated with the dark side of the edge, and negative for that part of the transition associated with the light side of the edge. Hence, the second derivative can be used to determine whether an edge pixel lies on the dark or light side of an edge.

Fig. 4 Edge detection by derivative operators: (a) light strips on a dark background; (b) dark strip on a light background. Note that the second derivative has a zero crossing at the location of each edge.

Segmentation

Gradient Operators

The gradient of an image *f*(*x, y*) at location (*x*,*y*) is given by :

$$\nabla \vec{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \qquad (3)$$

The gradient vector points in the direction of maximum rate of change.

In edge detection, an important quantity is the magnitude of $\nabla \vec{f}$, which is referred to simply as gradient, where

$$\nabla f = mag(\nabla \vec{f}) = \left[G_x^2 + G_y^2\right]^{1/2} \qquad (4)$$

mag $\left(\nabla \vec{f}\right)$ $\equiv$ maximum rate of increase of *f*(*x*,*y*) per unit distance in the direction of $\nabla \vec{f}$

Segmentation

Gradient Operators

Gradient is commonly approximated by the absolute values:

$$\nabla f \approx |G_x| + |G_y| \tag{5}$$

Direction of $\nabla \vec{f}$, at (*x*, *y*)

$$\alpha(x,y) = \tan^{-1}\left(\frac{G_y}{G_x}\right) \tag{6}$$

Where angle is measured with respect to the *x*-axis.

Derivatives may be implemented in digital form in several ways.

Segmentation

Gradient Operators – Sobel Operator

Sobel operators have the advantage of providing both a differencing and a smoothing effect

As differencing or derivatives in digital image processing enhance noise, the smoothing effect of Sobel operators is an attractive features.

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

a

-1	-2	-1
0	0	0
1	2	1

b

-1	0	1
-2	0	2
-1	0	1

c

Fig 5 (a) 3 x 3 image region; (b) Compute G_x at centre point of 3 x 3 region; (c) compute G_y at centre point of 3 x 3 region

Segmentation

Gradient Operators – Sobel Operator

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

a

-1	-2	-1
0	0	0
1	2	1

b

-1	0	1
-2	0	2
-1	0	1

c

Derivatives based on Sobel masks are

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \qquad (7)$$

$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \qquad (8)$$

Computation of the gradient at the central location of the mask then uses eqn (4) & (5) to give one value of the gradient.

The mask in then moved to the next location, and the process is repeated. We will obtain a gradient image which is of the same size as the original image.

Mask operations on the image border are implemented by using the appropriate partial neighbours. Fig. 6 shows an example of gradient images.

Segmentation

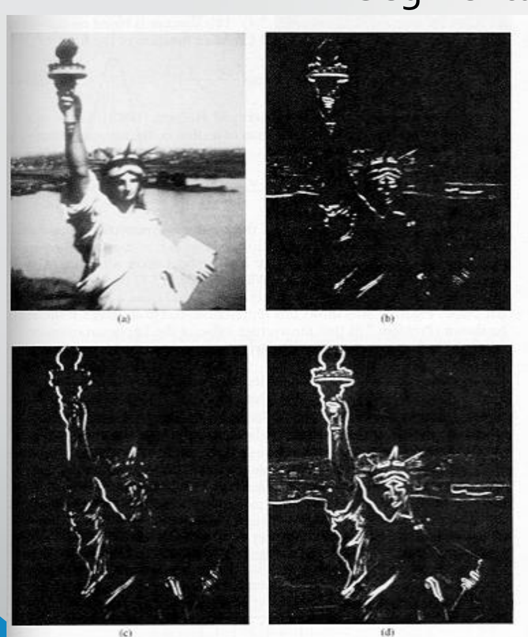


Fig. 6 (a) Original image; (b) result of applying the mask in Fig. 5(b) to obtain G_x ;

(c) result of applying the mask in Fig. 5(c) to obtain G_y ;

(d) complete gradient image obtain by using Eqn (5).

Segmentation -Laplacian

Laplacian

Lapacian of a 2-D function $f(x,y)$ is

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{9}$$

For a 3 x 3 region, the common implementation is

$$\nabla^2 f = 4z_5 - (z_2 + z_4 + z_6 + z_8) \tag{10}$$

The mask used in this case must have a positive central pixel, and the outer pixels must be negative. In addition, the sum of the coefficients must be zero.

0	-1	0
-1	4	-1
0	-1	0

Fig. 7 A mask used to calculate the Laplacian

Fig 7 is a spatial mask to compute the Laplacion of an image.

Segmentation -Laplacian

Laplacian responds to transition in intensity, but is seldom used in practice for edge detection, because

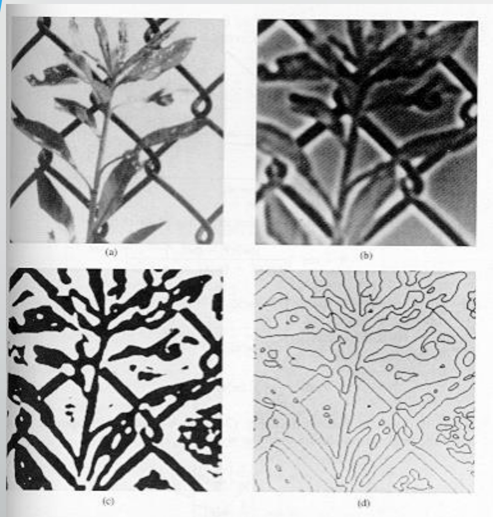
1. As a second order derivative, it is unacceptably sensitive to noise.
2. It produces double edges (See Fig 4).
3. It is unable to detect edge direction.

Laplacian usually play a secondary role of detector for establishing whether a pixel is on the dark or light side of an edge.

A more general use of Laplacian is in the finding the location of edges using its zero crossing property (Fig.4.)

Fig.8 shows an example of edge detection with Laplace Operator.

Segmentation -Laplacian



In Fig. 8(b), Black corresponds to the most negatives results, whilst white corresponds to the most positive. Note that mid-grays represent zero.

In Fig. 8(c), all negative results obtained in Fig. 8(b) are set to zero (black) and all positive results to white. Finally, in Fig. 8(d), zero crossings are identified as the boundary between black and white in Fig. 8(c).

Fig. 8 (a) Original image; (b) result of processing (a) with Laplace Operator; (c) results of making (b) binary to simplify detection of zero crossing; (d) zero-crossings.

Segmentation -Laplacian

Some Remarks on Gradient Operations in Edge Detection

1. They tend to work well in cases involving images with sharp intensity transitions and relatively low noise.
2. *Zero-crossings offer an alternative in cases when edges are blurry or when a high noise content is present.
3. They offer reliable edge location, and the smoothing properties of Laplace Operator to reduce the effect noise.
4. Computational complex & intensive.

Segmentation – Edge Linking

3. Edge Linking and Boundary Detection

The techniques described above detect intensity discontinuities. Ideally they should be able to detect pixels lying on the boundary between regions.

However, this set of pixels seldom characterises a boundary completely because of noise, breaks in the boundary from non-uniform illumination, and other effects that introduce spurious intensity discontinuities.

Thus, edge detection algorithms are usually followed by linking and other boundary detection procedures designed to assemble edge pixels into meaningful boundaries.

Segmentation – Edge Linking

3.1 Local Processing

This simple approach for linking edge points is to analyse the characteristics of pixels in a small neighbourhood (3 x 3, or 5 x 5, say) about every point in an image that has undergone edge detection.

All similar points are linked, forming a boundary of pixels that share some common properties.

Two principal properties are used:

- the strength of the response of the gradient operator used to produce the edge pixel, and
- the direction of the gradient.

Segmentation – Edge Linking

The first property is given by Eq. (4)

$$\nabla f = mag\left(\nabla \vec{f}\right) = \left[G_x^{\,2} + G_y^{\,2}\right]^{1/2} \tag{4}$$

or

$$\nabla f = |G_x| + |G_y| \tag{5}$$

Thus, an edge pixel at (*x'*, *y'*) and in the pre-defined neighbourhood of (*x,y*) is similar in magnitude to the pixel at (*x,y*), if

$$\left|\nabla f(x,y) - \nabla f(x',y')\right| \leq T \tag{13}$$

where *T* is a non-negative threshold.

The direction of the gradient vector is given by Eq. (6)

Segmentation – Edge Linking

The direction of the gradient vector is given by Eq. (6)

$$\alpha(x,y) = \tan^{-1}\left(\frac{G_y}{G_x}\right) \tag{6}$$

Then, an edge pixel at (*x',y'*) in the pre-defined neighbour of (*x,y*) has an angle similar to the pixel at (*x,y*), if

$$\left|\alpha(x,y) - \alpha(x',y')\right| < A \tag{14}$$

where *A* is the angle threshold.

Segmentation – Edge Linking

Therefore, a point in the pre-defined neighbourhood *f*(*x*,*y*) is linked to the pixel at (*x*,*y*) if both criteria, given in eqn (13) and (14) are satisfied. This process is repeated for every location in the image.

A record is kept for all linked points as the centre of neighbour is moved from pixel to pixel.

A simple book-keeping procedure is to assign a different gray level to each set of linked edge points / pixels.

Segmentation – Edge Linking

Fig 9(a) shows an image of the rear of a vehicle.
Objective: to find rectangles with sizes close to that of license plate.

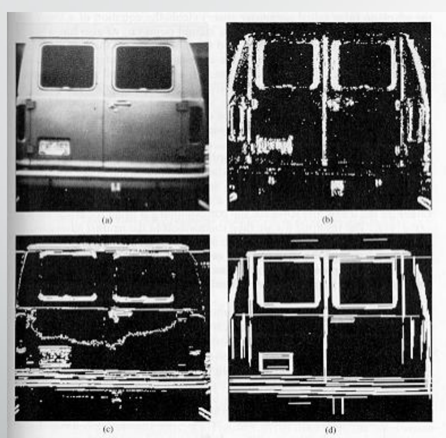


Fig. 9 (a) Input image; (b) $|G_x|$ component of the gradient; (c) $|G_y|$ component of the gradient; (d) result of edge linking

In Fig. 9(d), result is obtained by linking all points that simultaneously had

$$\left|\nabla \vec{f}\right| > 25$$

and

$$\left|\alpha(x,y) - \alpha(x',y')\right| < 15^0$$

Horizontal lines were obtained by sequentially applying these criteria to every row of Fig 9 (c), and a sequential scan of columns in Fig 9(b) yields the vertical lines. Further processing consisted of linking edge segments with small breaks, and deleting isolated segments.