References

Extract Threshold using Histogram

Amal Abbas Kadhim
College of Education-Mustansiriyah University

Abstract

In many vision application, it is useful to be able to separate out the regions of the image corresponding to objects in which we are interested, from the regions of the image that correspond to background [FIS 2000]. One of many ways used to separate the object from the background is edge detection methods. Thresholding often provides as easy way (technique) to perform the operation of edge detection. Because of its wide applicability to other areas of the digital image processing, quite a number of thresholding methods have been proposed over the years.

In this project a new algorithm is introduced in attempt to obtain the threshold value from the pixels intensity (histogram) of the gradient images. Two edge detectors are used to obtain the gradient images, one of them is traditional method (Sobel edge detector) and another has some complexity in its algorithm (Canny edge detector) in order to determine the efficiency of the adaptive algorithm.

الخلاصة

في التطبيقات الصاسية، من المفيد أن تكون هناك القدرة على فصل الأجسام المهمة عن المحيط الخارجي (Objects) واحدة من الطرق الكثيرة المستخدمة في عملية الفصل هذه هي كشف الحواف (Edge Detection) . تقنية آل
1 – Introduction

Edges characterize boundaries and are therefore a problem of fundamental importance in image processing. Automatic thresholding of the gray level values of an image is very useful in automatic analysis of more phologial images thresholding. Choosing the threshold as the value that maximizes the entropy of 1-dimensional (histogram) of an image, one might be able to separate, effectively, the desired objects from the background. Not all images can be neatly segmented into foreground and background using simple thresholding, whether or not an image can be correctly segmented this way can be determined by looking at an intensity histogram of the image. If it's possible to separate out the foreground of an image on the basis of pixel intensity, then the intensity of pixels within the foreground object must be distinctly different from the intensity of pixels within the background. In this case, we expect to see a distinct peak in the histogram corresponding to foreground objects such that threshold can be chosen to isolate this peak accordingly. If such peak does not exist, then it is unlikely that simple thresholding will produce a good segmentation [FIS 2000].

Some of the literature works concerned with the using histogram to extract the edges in images are:-

Katz, in 1965, points out that since the pixels in the neighborhood of an edge have higher edge value; the gray level histogram for the pixels should have a single peak at gray level between the object and background gray levels. This gray level is, therefore, a suitable choice of the threshold value [SAH 88].
Sahoo, in 1987, present a survey of thresholding techniques and update the earlier survey work by Weszka and Fu and Mu to attempt to evaluate the performance of some automatic global thresholding methods using the criterion function such as uniformity and shape measures. The evaluation is based on some real world images [SAH 88].

Abutaleb, in 1988, show that by choosing the threshold as the value that maximizes in the entropy of the 1 - dimensional histogram of an image, one might be able to separate, effectively, the desired objects from the background. The approach does not take into consideration the spatial correlation between the pixels in an image. Thus, the performance might degrade rapidly as the spatial interaction between pixels becomes more dominant than the gray-level values [ABU 89].

2 - The edge detectors

Edges in images are areas with strong intensity contrasts – a jump in intensity from one pixel to the next. Edge detecting an image significantly reduces the amount of data and filters out useless information, while preserving the important structural properties in an image. In this paper two edge detectors are used:

2 - 1 Sobel edge detector

Sobel edge detector is a great algorithm to learn the basics of edge detection. Two 3x3 convolution masks in figure (1) are applied to each pixel, one with a horizontal trend and one with a vertical trend. The result of each convolution is treated a vector representing the edge through the current pixel. If the magnitude of the sum of these two orthogonal vectors is greater than some user-specified threshold, the pixel is marked in black as an edge. Otherwise, the pixel is set to white [BRO01].

![Sobel operator convolution masks](image)

Fig (1): Sobel operator convolution masks
2 - Canny edge detector

The Canny edge detection algorithm is known to many as the optimal edge detector. He presents a list of criteria to improve current methods of edge detection. The first and most obvious is low error rate. It is important that edges occurring in images should not be missed and that there be no responses to non-edges. The second criterion is that the edge points be well localized. In other words, the distance between the edge pixels as found by the detector and the actual edge is to be at a minimum. A third criterion is to have only one response to a single edge. This was implemented because the first 2 were not substantial enough to completely eliminate the possibility of multiple responses to an edge [GRE02]. The steps of Canny operator are:-[WAN04]

1 - Smooth the image with a Gaussian filter to reduce noise and unwanted details and textures.

\[ g(m, n) = G_m(m, n) + f(m, n) \quad \ldots \ldots (1) \]

where

\[ G_m = \frac{1}{\sqrt{2\pi}\sigma^2} e^{-\frac{m^2 + n^2}{2\sigma^2}} \quad \ldots \ldots (2) \]

2 - Compute gradient of \( g(m, n) \) using any of the gradient operators (Roberts, Sobel, Prewitt, etc) to get:

\[ M(m, n) = \sqrt{g_n^2(m, n) + g_n^2(m, n)} \quad \ldots \ldots (3) \]

and

\[ \theta(m, n) = \tan^{-1}\left[ g_n(m, n) / g_n(m, n) \right] \quad \ldots \ldots (4) \]

3 - Threshold M:
\[ M_T(m,n) = \begin{cases} M(m,n) & \text{if } M(m,n) > T \\ 0 & \text{otherwise} \end{cases} \] (5)

Where \( T \) is so chosen that all edge elements are kept while most of the noise is suppressed.

4 - Suppress non-maxima pixels in the edges in \( M_T \) obtained above to thin the edge ridges (as the edges might have been broadened in step 1). To do so, check to see whether each non-zero \( M_T(m,n) \) is greater than its two neighbors along the gradient direction \( \Theta(m,n) \). If so, keep \( M_T(m,n) \) unchanged, otherwise, set it to 0.

5 - Threshold the previous result by two different thresholds (upper threshold) \( T_u \) and (lower threshold) \( T_l \) (where \( T_l < T_u \)) to obtain the result images.

3 - Thresholding Using Histogram

Thresholding is a widely used tool in image segmentation where is interested in identifying the different homogeneous components of the image. It is useful in discriminating objects from background in many scenes [ABU 89].

The term histogram has several meanings. Originally a graphical tool developed by statisticians to visualize frequency distributions, it has come to have a very specific meaning when used in the context of digital images [SAC96], so an image histogram is a chart that shows the distribution of intensities in an indexed or intensity images.

There are two types of histogram: bilevel and multilevel. In bilevel as in figure (2.a) the gray level distribution (histogram) is usually assumed to have one valley between two peaks; each represents the gray level concentration of one segment. The objective is to locate, effectively, the bottom of the valley which, hopefully, best separates the two groups or segments. In the multilevel, it is assumed that there are several components (segments) in the image, each of the homogeneous value of gray level. One then attempts to locate the values of the thresholds that can separate their components, so the threshold segmentation requires two thresholds \( T_1 \) and \( T_2 \) as in figure (2.b). In figure (2.c) the two peaks of a distribution have run together and so it almost certainty not possible to successfully segment the image using single
threshold. An adaptive algorithm that obtained the threshold value from the histogram of the gradient image is introduced:

![Typical histogram with suitable choice of threshold](image)

**Fig (2): Typical histogram with suitable choice of threshold**

1. Calculate the gradient images by applying the masks of the edge detectors on original images (Sobel and Canny edge detectors).
2. Determine and draw the histogram of the gradient images.
3. Applying 3x3 mask on the gradient image and calculate the average for each mask.
4. Count the number of gradient values above the average value of the center pixel for each mask applied in step 2 (n).
5. Divide n obtained in step 4 by the size of the image to calculate the internal threshold (th).
6. Determine the histogram of the maximum value in the gradient image:
   \[ i = 255, \quad \text{Acm} = \text{his}(i), \quad T = i / \text{Acm} \]
7. Compare T with internal threshold obtained in step 5 (th), while T < th then add his (i-1) to Acm.
8. When the comparison is complete then (i) will be the threshold used in edge detection operation.
4 – Experimental Results

The results of applying the Sobel and Canny edge detectors on selected images are introduced by using two methods of thresholding:

1 – Manual threshold

In this case the trial and error mechanism is used for choosing the suitable manual threshold value.

2 – Threshold from histogram

In this case the threshold value is obtained from the histogram of the gradient images using the algorithm that illustrate in section 3. The results show that the threshold value obtained from the histogram is close to manual threshold value, and to reach to the real value, the threshold value is multiplied by suitable factor.

4 – 1 Results of Sobel edge detector

The results of applying Sobel edge detector on test images is shown in figure 3. Where most of the edges have been detected. The histogram of the gradient images does not have distinct peaks and there is no valley can be chosen as threshold value. By applying the adaptive algorithm to choose the threshold value from the histogram of the gradient images it is shown that the threshold value need a factor does not exceed 2 to reach to the manual threshold value.

4 – 2 Results of Canny edge detector

The result of applying Canny edge detector is shown in figure (4), where the edges are detected with some thickness. It is important to show that the threshold value obtained from the histogram is very close to lower threshold (TL), so the upper threshold value (Tu) is obtained by applying the (TL) by 2.
Fig (3) The result of applying the Sobel edge detector
5 – Conclusion

From the results obtained of applying the adaptive algorithm to extract the threshold value from the histogram of the gradient images, some point can be derived:
1 – The threshold value obtained from the histogram is close to manual threshold and the factor doesn’t exceed (2) in testing images.

2 – In Canny edge detector, the value of threshold obtained from the histogram is reach to TL and the Tu is reached by multiplying TL with (2).

6 – Future Work

1 – The adaptive algorithm that used to obtain the threshold value used in edge detection from the histogram can improved to cancel the factor value that entered from the user to calculate the threshold value from the histogram automatically.

2 – The adaptive algorithm can applied on color images to obtain the threshold value in edge detection in color image.

References

- [ABU 89]

- [BRO001]

- [FIS 2000]

- [GRE 02]

- [SAC 96]

- [SAH 88]

[ WAN 04]