A Study into Entropy-based Thresholding for Image Edge Detection*

Ramiro Liscano
Autonomous Systems Laboratory
National Research Council of Canada
Ottawa, Ont., Canada K1A 0R6
e-mail: liscano@iit.nrc.ca

Andrew K. C. Wong
Department of Systems Design
University of Waterloo
Waterloo, Ont. N2K 3G1
e-mail: akcwong@watnow.uwaterloo.ca

Abstract

Automated thresholding approaches have normally been applied to gray-level intensity images to differentiate between objects and the background in the image. This paper investigates the use of an entropy-based thresholding approach for determining a reasonable threshold value for an intensity gradient in an edge tracking algorithm and a threshold value for the lengths of edges extracted from an image. The histograms of the intensity gradient of an edge and the lengths of edges generally peak very quickly at low values and quickly drop as their values increase. The entropy-based thresholding technique is adequate for determining a reasonable threshold value for these type of histograms, particularly since it computes the point at which the information content of the two sides of the histogram is a maximum. The paper also demonstrates the importance of reapplying the threshold determination algorithm on different parts of the image, since the threshold value is relative to the distribution in a region of interest. The effects of sparse data on the computation of the threshold are investigated and an example is presented demonstrated the strong impact that sparse data can have.

1 Introduction

In the mid 1980's there was a surge in developing thresholding techniques to be applied to intensity images for the purpose of extracting objects in the image from the background. The use of these techniques to applications beyond their conventional gray-level thresholding is not abundant in the literature even though it is fairly common in computer vision to encounter the use of thresholding to reduce the number of features to an amount computationally manageable.

There exists many methods for computing thresholds in gray level images and the reader is referred to surveys by Sahoo et al. [1] and by Weszka [2]. In this paper an entropy-based method developed by Kapur et al. [3] will be evaluated in determining threshold values for the minimum intensity gradient in an edge detecting algorithm and a threshold value for the length of edges in an image. The original work by Kapur et al. focused primarily on the computation of threshold values in image intensity while this paper applied the technique to edge detection and edge lengths. The motivation comes from the desire to reduce the computational costs associated with feature grouping for perceptual organization which can be in the order of $\mathcal{O}(n^2)$, where n is the number of features extracted from the image. Reducing the number of features extracted from an image will also reduce the computational complexity for object recognition, which at worst can be in the order of $\mathcal{O}(n^2m^2)$ where m is the number of features in the model. The threshold algorithm is at worst linear in complexity relative to the number of features so the savings can be substantial if the number of features is large.

This paper utilizes an entropy-based thresholding technique which computes a threshold value based on the information content of the histogram of the features in an image. Automating the thresholding procedure is of particular importance in situations where the measures computed for the extraction of features from an image are difficult to visualize. A good example , is i n the computation of a reasonable gradient threshold value for an edge detector. The actual gradient values have little physical significance to the user of the algorithm and there is a

*NRC Number 38362.

difficulty in manually determining a threshold value since the histogram of the features has not any clear valleys which could serve as a reasonable threshold value. Using an entropy approach offers an "objective" manner in determining the threshold value since the computation of the threshold is based on the information content of the data.

Another important issue in thresholding operations is that in most situations the threshold value is relative to the values in the image and not an absolute value. The consequence of this is that threshold values will change based on the region of focus within the image so that one single threshold operation will not in general extract all the desired features. The issue of how to choose those particular regions of interest will not be discussed in this paper.

This article is divided in the following manner; Section 2 gives an outline of Kapur's [3] entropy-based thresholding approach. Section 3 shows two applications of the method in determining a threshold value for the gradient value of an edge tracking algorithm and the length of a set of edges extracted from an image. Section 4 analyses the results from the test cases discussed in section 3. The final section, section 5, summarizes the paper and makes some recommendations on the use this algorithm.

2 Entropy-based thresholding algorithm

This algorithm as developed by Kapur et al. [3] is based on Shannon's information entropy measure and is designed to maximize the amount of information between the two parts of a histogram as defined by a threshold point in the histogram. In this paper, the method has been applied on domains beyond the conventional intensity thresholding operation of which it was originally designed for. There is still little evidence that thresholding techniques based on Shannon's entropy measure are better suited than other non-bimodal thresholding approaches except that intuitively there seems to be an attraction to maximizing the information content between the two sections of data created by the threshold operation.

The procedure commences by first creating a histogram, partitioned into n parts, that ranges from the minimum to the maximum value of the extracted feature. This first operation maps the features into a dimensionless probability space allowing us to ignore for most part of the operation any physical dimensions in the data. The concept of the entropy-based algorithm is on the separation of the histogram into two sections, A and B, where the histogram of section A is from 1 to s and the rest is in section B. The value associated with the histogram location of s is then considered as the threshold value.

The two sections have the following probability distributions,

$$A : \frac{p_1}{P_s}, \frac{p_2}{P_s}, \ldots, \frac{p_s}{P_s}, \tag{1}$$

$$B : \frac{p_{s+1}}{1-P_s}, \frac{p_{s+2}}{1-P_s}, \ldots, \frac{p_n}{1-P_s}, \tag{2}$$

where the values $p_i : i = 1, \ldots, n$ are the probabilities associated with the partitions in the histogram and $P_s = \sum_{i=1}^{n} p_i$.

The Shannon entropies associated with these sections are computed by applying Shannon's entropy equation for a single observation X,

$$H(X) = \sum_{k}^{n} -P_k \ln P_k \tag{3}$$

to sections A and B in the histogram. These entropies are the following,

$$\begin{aligned} H(A) &= -\sum_{i=1}^{s} \frac{p_i}{P_s} \ln \frac{p_i}{P_s}, \\ &= \ln P_s + \frac{H_s}{P_s}. \end{aligned} \tag{4}$$

and,

$$\begin{aligned} H(B) &= -\sum_{i=s+1}^{n} \frac{p_i}{1-P_s} \ln \frac{p_i}{1-P_s}, \\ &= \ln(1-P_s) + \frac{H_n - H_s}{1-P_s}, \end{aligned} \tag{5}$$

where $H_n = -\sum_{i=1}^{n} p_i \ln p_i$ and is defined as the entropy over the whole histogram, and $H_s = -\sum_{i=1}^{s} p_i \ln p_i$ and is the entropy of the section in the histogram from the start to the threshold location s.

Defining $\Psi(s)$ as the sum of $H(A)$ and $H(B)$,

$$\Psi(s) = \ln P_s(1-P_s) + \frac{H_s}{P_s} + \frac{H_n - H_s}{1-P_s}. \tag{6}$$

The threshold value is then defined as the location in the histogram where $\Psi(s)$ is the maximum. This results in a simple but elegant approach for determining a threshold value for the examples that follow in the next section.

3 Examples

Equation 6 can be applied to determine a threshold value of any measure of a feature. In this section two examples are given. The first determines a threshold value using an intensity gradient measure from an image and the second computes a threshold for the length of edges extracted from an image. For both examples the histogram consisted of 64 partitions over the minimum and maximum values of the measure. Once the histogram is computed, equation 6 is applied to the probability values in the histogram using values of $s = 1$ to $s = 64$ and recording when $\Psi(s)$ is a maximum. The value associated with the histogram region s is then taken as the threshold value.

3.1 Intensity gradient threshold

This thresholding technique was applied to an intensity image to determine a reasonable threshold in intensity gradient to be used in an edge tracking algorithm. The edge tracking algorithm scans along the horizontal direction of an image looking for a location where the intensity gradient is greater than a certain specified threshold value G_t. This location then becomes a viable starting point to track an edge. Once all the pixels have been checked along the horizontal scan the same procedure is repeated along the vertical scan of the image to detect any edges whose maximum gradient lies along the vertical direction and do not already belong to another edge.

This algorithm used the following gradient operator,

$$G_r(i) = -I(i-2) - 2 * I(i-1) + 2 * I(i+1) + I(i+2), \quad (7)$$

where the value of i is the current pixel location and the function $I(i)$ computes the intensity value at pixel location i. When applied in the horizontal direction then pixels $i-a$ and $i+a$ correspond respectively to pixels to the left and right an amount a from pixel i, and when applied in the vertical direction, pixels $i-a$ and $i+a$ correspond respectively to pixels above and below by an amount a. This gradient operator has been used successfully in tracking edges for the purpose of examining a new invariant measure for curve detection based on perceptual organization [5].

The entropy-based thresholding technique is ideal in determining G_t for this situation because it is difficult to attach a physical feeling for the gradient value $G_r(i)$ from equation 7, and this threshold

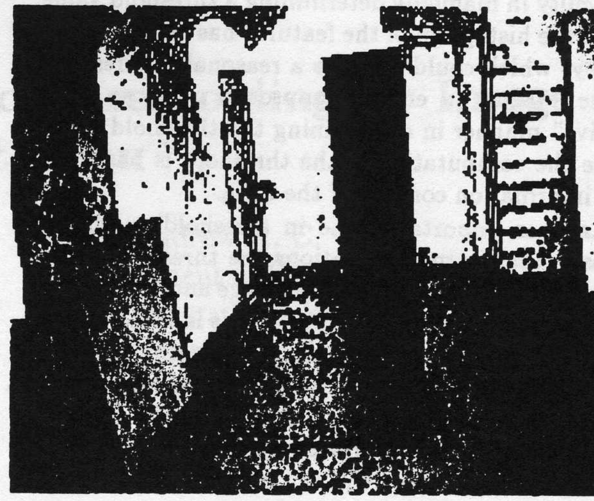


Figure 1: Image of detected edges using $G_t = 10$

value should be relative to the intensity values in the image.

Figure 1 is an image of the PAMI lab at the University of Waterloo displaying the edges that were extracted using equation 7 along with a small gradient value (in this case 10) as the minimum gradient value of an edge. Using a gradient value of zero would have resulted in edges being detected for any pixel in the image.

Figure 2(a) shows the histogram of the gradient values in the image. Notice the single peak near the start of the histogram and the relatively small values after that. Also in the histogram is a pointer to the threshold value computed by applying equation 6 to the histogram. Figure 2(b) shows the edges extracted from the image using a threshold gradient value of 120 which was approximately the threshold value computed from the histogram.

The edge tracking procedure used was developed by Gao [6] and uses the same intensity gradient equation as that shown in equation 7. In that approach the threshold gradient value G_t was used only for determining the commencement of an edge to be tracked and a lower threshold value was used to terminate the tracking operation. It is therefore possible to have edges that contain pixel values with gradient values below the threshold value of G_t. The effect of the thresholding operation was to reduce the number of edges from about 4,200,000 to 108.

It is desirable to extract some other edges from the image. In particular some of the edges near the column to the left of the image and of the board resting on the floor and leaning against the col-

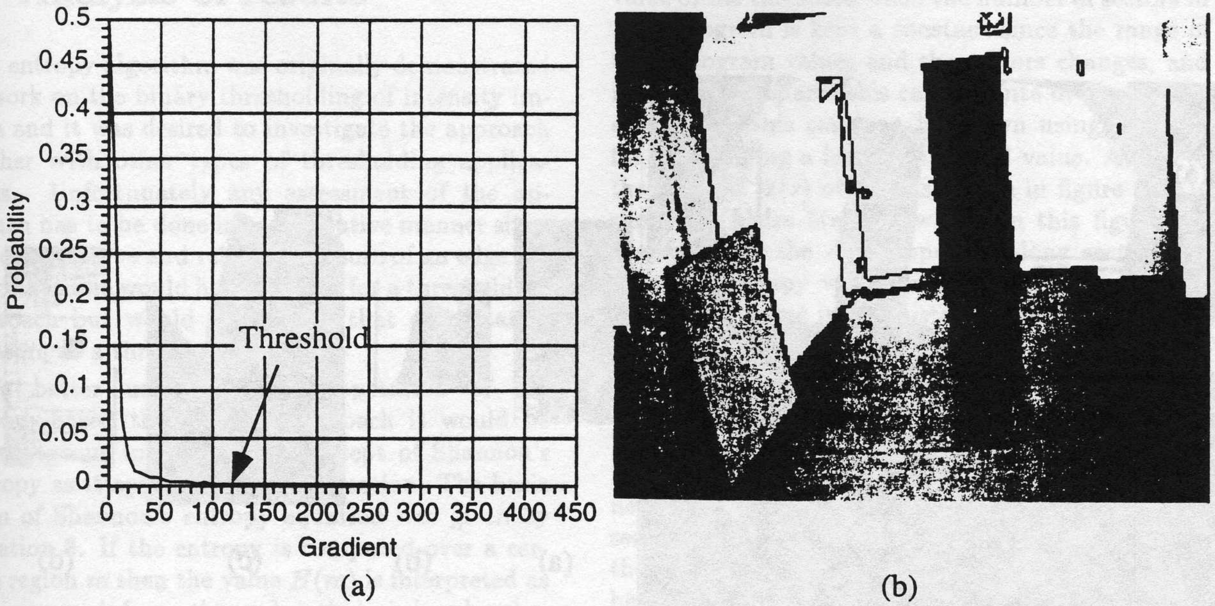


Figure 2: Histogram of the gradient values (a), and extracted edges using $G_t = 120$.

umn. In this case it is more advantageous to repeat the procedure on a selected part of the image. That selection may come from information about the already extracted features, for example corner junctions are important to investigate further, or may be an overall operation on the image using windows of particular sizes. The latter approach would more likely lead to a numerous number of edges particularly if small sized windows are used. For these examples we have, using our human judgment, decided that the algorithm would be applied to a few locations where there are some edge junctions. These areas are shown as rectangles in the full image in figure 3 and have been labeled with the letters (a) through (d). The results of applying the threshold operator, computed using equation 6, to the smaller sections of the image are shown in the figures 3(a), 3(b), 3(c), and 3(d).

These results are fairly promising in that from the sub-sections of the larger image it was possible to compute threshold values that resulted in again most of the noisy edges removed and some of the primary edges remaining. It also demonstrates the importance of a form of re-iteration of the procedure in particular to areas of the image where there is a desire to verify the existence of other edges. Figure 3 (d) is part of the image which has a fairly uniform intensity except for a light switch on the wall. It is interesting to note that for this example it was not possible to extract the faint edges created from the bricks that form the wall, instead the threshold value was influenced by the higher contrast light switch which has boundaries with gradient values greater than the threshold value.

3.2 Edge length threshold

Another approach in minimizing the number of edges that are extracted from an image is to consider a histogram of the lengths of the edges. In this case we have a better physical sense to the length of an edge and it could be easy to specify a threshold based on an absolute value. The problem considered here is one where it is not desired to specify any absolute values for the length of an edge so that it becomes necessary to compute one from the sensory data.

Figure 4(a) shows a plot of the histogram of the lengths of the edges shown in figure 1. Applying the entropy-based thresholding technique to this histogram resulted in a length threshold of approximately 100. The results of applying this threshold value to the image are shown in figure 4(b)

In this example the reduction in the number of edges was from 5625 edges to 10. The savings in performing a grouping operation, like checking for parallelism or symmetry, is now reduced from at worst 5625^2 operations to 10^2.

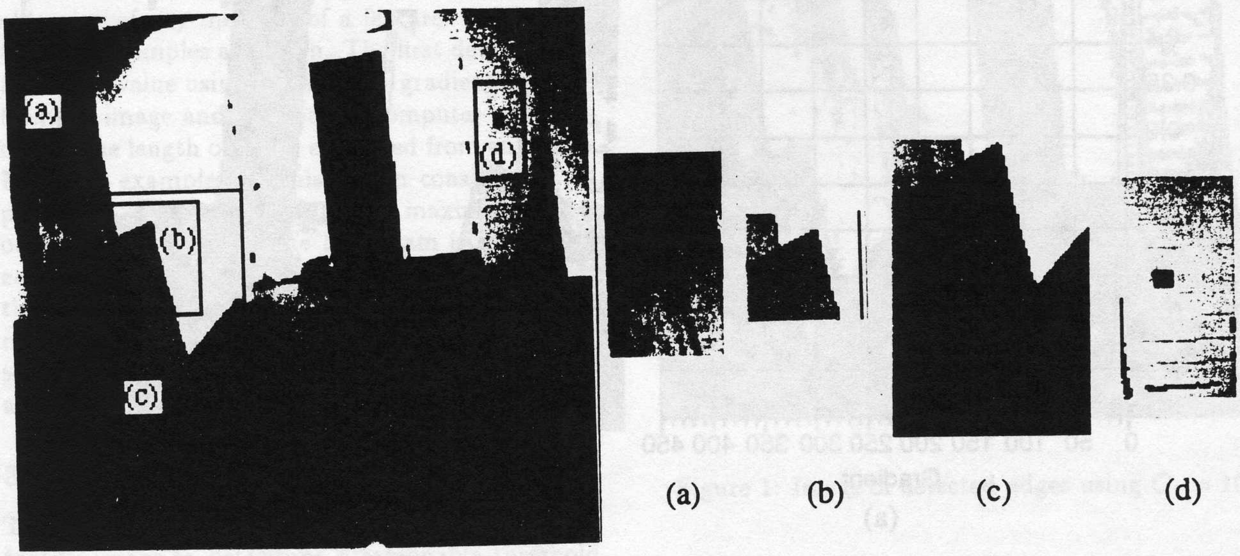


Figure 3: Tracked edges in the smaller parts of the image.

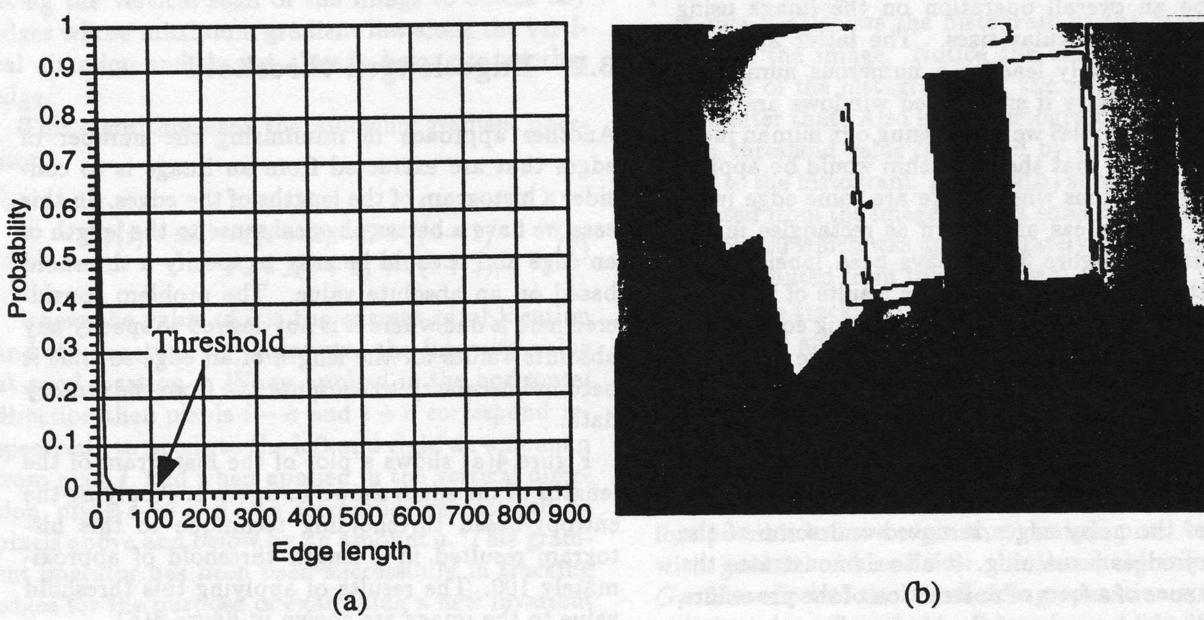


Figure 4: A histogram of edge lengths (a) and edges greater in length than 100 (b)

4 Analysis of results

The entropy algorithm was originally demonstrated to work on the binary thresholding of intensity images and it was desired to investigate the approach further with other types of thresholding applications. Unfortunately any assessment of the approach has to be done in a qualitative manner since if a quantitative and reliable measure of an edge existed then one would have no need for a thresholding approach but would simply use that quantitative measure as a threshold.

To better understand the operation of the entropy-based thresholding approach it would be advantageous to discuss the concept of Shannon's entropy as it applies to this application. The basic form of Shannon's entropy equation was given by equation 3. If the entropy is computed over a certain region m then the value $H(m)$ is interpreted as the average information value that each subregion within m supplies to the whole region m.

The threshold value tries to determine a location among the histogram of the data where the entropy value of the distribution of sector A plus the entropy value of the distribution of sector B are a maximum. It is trying to partition the histogram so that the expected amount of information in sector A in addition to the expected amount of information in sector B is a maximum. The equation unfortunately is very complicated to analyze and it is difficult to come up with a general feeling for how it behaves relative to the values in the histogram. For the specific cases explored in this paper the histograms have large peak values close to the low values of the histogram, as shown in figures 2(a) and 4(a), and quickly drop down to smaller probability values past that point. In both these cases threshold values have been computed past the peak and closer to the peak than near the center. Further analysis and experimentation are required to achieve this general feeling.

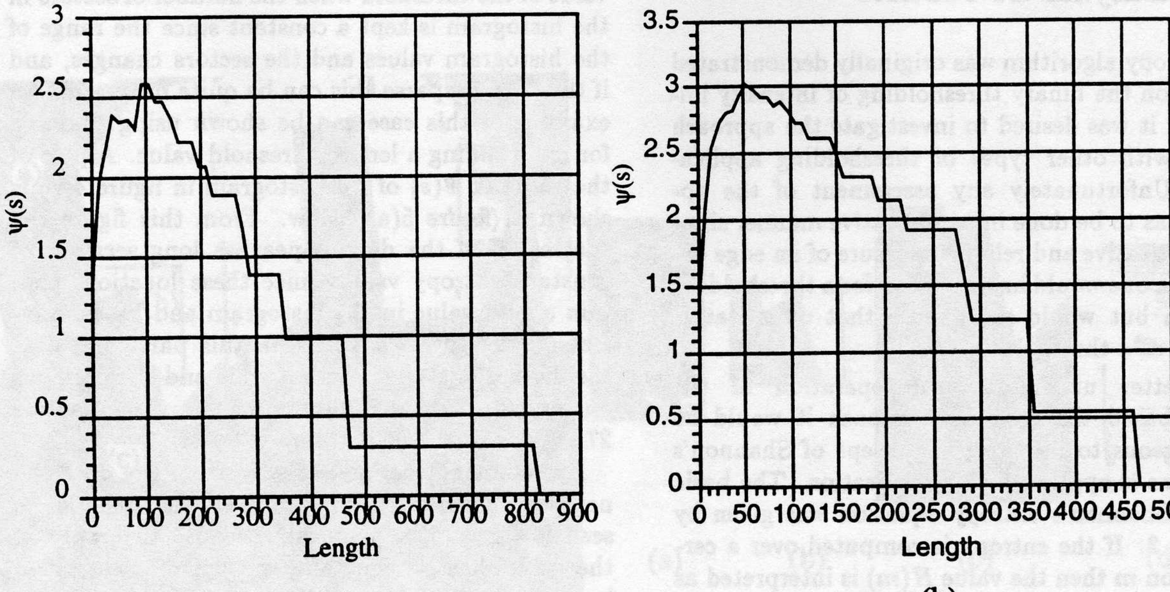
Perhaps as important is the sensitivity of the algorithm to outliers in the data. Since the values of the features are being mapped onto probability space, and since the probability of the outlier data is small, the calculations of the threshold location in the histogram is not very sensitive to outliers in the data. In a similar direction of thought, any locations in the histogram with a zero count will not affect the overall entropy value and therefore it is not necessary to make special cases for them. This is of particular concern when the data is sparse since there can exist large sections of the histogram with no values. Outliers in the data though do affect the value of the threshold when the number of sectors in the histogram is kept a constant since the range of the histogram values and the sectors changes, and if the data is sparse this can be quite dramatic. An example of this case can be shown using the case for determining a length threshold value. A plot of the value of $\Psi(s)$ of the histogram in figure 4(a) is shown in figure 5(a) below. From this figure the sparseness of the data appear as long sections of constant entropy values since these locations contain a zero value in the histogram and have no effect on the entropy value. In this particular case the distance between the last edge and the previous one is approximately 360 units and spanned across 27 regions in the histogram.

By removing the longest edge from the data the new histogram takes on a radically new shape, as seen in figure 5(b) primarily because the range of the values has changed and the size of each region becomes smaller.

In the case where the last edge was kept, the threshold value was 120 and resulted in 10 edges with a length greater than 120, while in the second case, without the last edge, the threshold value was about 50 and resulted in 30 edges of length greater than 50. This shows that the method is sensitive to outliers in the data the outcome being that small changes in the data can result in significant changes in threshold values for sparse histograms. This is compounded by the problem that practically in this example one does not really consider long edges as outliers so it is not recommended that they be removed. One solution to the problem is to consider histograms with fixed partition sizes instead of fixed number of partitions, the removal of the outlier in this case will only reduce the total number of partitions but have little effect on the probability values of the other partitions. If it is necessary to have a fixed number of partitions then one may consider the use of variable size partitions and implement a maximum-entropy partitioning approach as described by Chieu et al. [4] to determine the boundaries of the partitions. The maximum-entropy partitioning approach tries to equalize the probabilities in each partition of the histogram and therefore is less sensitive to the removal of an outlier.

5 Concluding remarks

This paper demonstrated the use of an entropy-based thresholding technique on domains beyond the conventional intensity thresholding operation of which it was originally designed for. Through this investigation further insight into the advantages and



(a) (b)

Figure 5: $\Psi(s)$ for the histogram of edge lengths: (a) All edges; (b) longest edge removed

limitations of the approach have surfaced. The particular advantage of this method is that it can be applied to histograms that contain no definite valleys since it is based more on the maximization of the information content between the two sections defined by the threshold value. The histograms of the examples discussed were typical of edge detection approaches and the approach did compute threshold removing a large majority of short edges or edges with small gradient values. It was also demonstrated the importance of reapplying the threshold determination algorithm on different parts of the image, since this results in threshold values appropriate to that region in the image. A major concern of this approach is in the use of sparse data so it is necessary to analyze the histogram before computing a threshold value and determining if the sparse data are values that should be kept in the histogram or discarded before the operation. An interesting direction to investigate is to combine the results of both the image gradient thresholding and the edge length thresholding for edge extraction. The gradient threshold is more appropriate for determining a threshold for a particular region in the image, while the length threshold can be applied on the full image to extract the longer edges.

References

[1] P. K. Sahoo, S. Soltani, and A. K. C. Wong, "A survey of thresholding techniques", *Computer Vision, Graphics, and Image Processing*, vol. 41 (1988), pp. 233-260.

[2] J. S. Weszka, "A survey of threshold selection techniques", *Computer Graphics and Image Processing*, vol. 7 (1978), pp. 259-265.

[3] J. N. Kapur, P. K. Sahoo, and A. K. C. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram", *Computer Vision, Graphics, and Image Processing*, vol. 29 (1985), pp. 273-285.

[4] D. K. Y. Chiu, A. K. C. Wong, and B. Cheung, "Information discovery through hierarchical maximum entropy discretization and synthesis", *Knowledge Discovery in Data Bases*, (1991), pp. 125-140.

[5] Q. Gao and A. K. C. Wong, "Curve detection based on perceptual organization.", *Pattern Recognition*, vol. 26, no. 7 (1993), pp. 1039-1046.

[6] Q. Gao, "Object recognition based on visual perception.", *PhD Thesis*, University of Waterloo (1993).