Unimodal Thresholding

Paul L. Rosin
Department of Information Systems and Computing
Brunel University
UK
Paul.Rosin@brunel.ac.uk

Abstract

Most thresholding algorithms have difficulties processing images with unimodal distributions. In this paper an algorithm, based on finding a corner in the histogram plot, is proposed that is capable of performing bilevel thresholding of such images. Its effectiveness is demonstrated on synthetic data as well as a variety of real data, showing its application to edges, difference images, optic flow, texture difference images, polygonal approximation of curves, and image segmentation.

1 Introduction

Over the years many image thresholding techniques have been developed [16], and considerable research continues nowadays [3, 4, 6]. The reason for this longterm, ongoing effort is that none of the methods are capable of optimal performance under all conditions. Although techniques based on many different principles are available they all operate under certain implicit or explicit assumptions. These assumptions effectively form restrictions on the successful operation of the thresholding algorithms. Examples of requirements for bilevel thresholding are that there should be two distinct modes in the intensity histogram, the peaks should not be too disimilar in size, and they should be roughly Normal. Since different computer vision applications produce different types of images we can expect the success of any thresholding algorithm to vary according to how well its expected operating conditions are met.

This paper looks at the case where bilevel thresholding is desired even though the image histogram may only contain one obvious peak. Although performing thresholding might not seem appropriate in such cases there are in fact many instances when it is required. Take for example edge detection, which typically produces many low magnitude edgels corresponding to non-edges. The true edges produce a wide range of edgel magnitudes that will often just create a fat tail on the non-edge peak in the edge histogram rather than generate a distinct peak of their own. Another example is change detection carried out by differencing pairs of images. In applications such as surveillance the amount of change will be extremely small if a wide field of view is imaged and only a small, distant object has moved. Thus the histogram could contain one enormous peak (no-change) and one tiny peak (change).

Such applications will cause difficulties for most existing thresholding algorithms, and there has been relatively little work in such areas [14, 15]. In contrast, we present in this paper a technique that is specifically intended to cope with essentially unimodal distributions rather than the more usual bimodal or multimodal distributions. Section 2 describes the algorithm and analyses its performance on a variety of types of synthetic data. In section 3 it is applied to thresholding a range of types of real data: edges, difference images, optic flow, texture difference images, polygonal approximation of curves, and image segmentation. Both the theoretical and experimental analyses demonstrate the benefits of the new algorithm for such data.

There is a wide range of thresholding techniques [16]. For instance, Otsu [11] minimises the ratio of the between class variance to the within class variance of the two classes. More recently many algorithms based on maximising some measure of entropy have been developed [2, 9]. As a last example are those based on the shape of the histogram, looking for concavities as suitable threshold points. Rosenfeld and de la Torre's [13] work in this vein is similar to ours, although in contrast to our approach they specifically attempt to eliminate unbalanced concavities.

2 Proposed Method

The proposed bilevel thresholding algorithm is extremely simple. It assumes that there is one dominant population in the image that produces one main peak located at the lower end of the histogram relative to the secondary population. This latter class may or may not produce a discernible peak, but needs to be reasonably well separated from the large peak to avoid being swamped by it. A straight line is drawn from the peak to the high end of the histogram. More precisely, the line starts at the largest bin and finishes at the first empty bin of the histogram following the last filled bin. If the i'th entry of the histogram is written as H_i then the line $(x_s, y_s) \rightarrow (x_f, y_f)$ is defined as $(\arg\max_i H_i, \max_i H_i) \rightarrow (\max_{H_i=0 \text{ and } H_{i-1}\neq 0} i, 0)$. The threshold point is selected as the histogram index i that maximises the perpendicular distance between the line and the point (i, H_i); see figure 2. This is effectively an application of a single step of the standard recursive subdivision method for determining the polygonal approximation of a curve [12].

Several assumptions have been made. First, that the large class has lower intensity values than the small class.[1] While it is not always realistic to expect this, in many cases the relative ordering is known. For instance, in the examples of edge and change thresholding given in the introduction the large peak corresponds to noise and is therefore at the bottom end of the histogram.

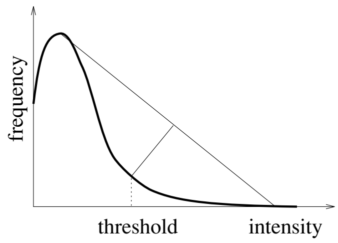


Figure 1

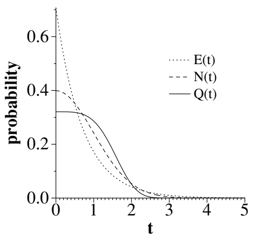


Figure 2

Figure 1: Procedure for determining threshold from intensity histogram;
Figure 2: Different distributions used as simple models in the analysis of the thresholding algorithm

The second assumption is that the main peak has a detectable corner at its base which corresponds to a suitable threshold point. To check the validity of this we shall first analyse the performance of the technique on some simple analytic distributions (see figure 2). First we look at modelling the large peak by a Normal distribution with zero mean and standard deviation σ

$$G(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-t^2}{2\sigma^2}}. \tag{1}$$

For simplicity we shall assume that the secondary peak is small enough and sufficiently well separated such that we can ignore it. If the threshold is chosen at position t then we can calculate the probability of error in classification (i.e. how much of the distribution is above the threshold) as

$$P(t) = \frac{\int_t^\infty G(x)dx}{\int_0^\infty G(x)dx} = \frac{1}{2}\left(1 - \operatorname{erf}\frac{t}{\sqrt{2}\sigma}\right) \tag{2}$$

where only the positive half of the distribution (i.e. positive intensities) is considered. While the Normal is a common model for the mode we shall also investigate the effects of other shaped distributions with peaks which are more or less sharp (i.e. leptokurtic and platykurtic). Thus, in a similar manner we model the peak by an exponential

$$E(t) = \frac{1}{\sqrt{2}\sigma} e^{\frac{-\sqrt{2}|t|}{\sigma}} \tag{3}$$

with a probability of error

$$P(t) = e^{\frac{-\sqrt{2}t}{\sigma}} \tag{4}$$

and as an exponential of the fourth power, giving

$$Q(t) = \frac{1}{\sigma\Gamma\left(\frac{1}{4}\right)\sqrt{C}} e^{-\frac{t^4}{16C^2\sigma^4}} \tag{5}$$

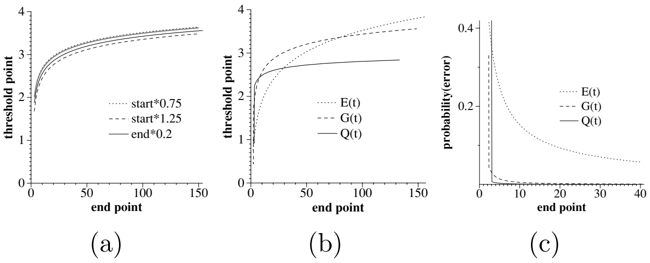
where

$$C = \frac{\Gamma\left(\frac{5}{4}\right)}{\Gamma\left(\frac{3}{4}\right)} \tag{6}$$

and the probability of error is

$$P(t) = \frac{\Gamma\left(\frac{1}{4}, \frac{t^4}{16C^2\sigma^4}\right)}{4\Gamma\left(\frac{5}{4}\right)} \approx 0.2758\Gamma\left(\frac{1}{4}, 0.1142\frac{x^4}{\sigma^4}\right). \tag{7}$$

To avoid confusion we emphasise that for uniformity all the distributions have been defined as symmetric about $t = 0$ with standard deviation σ. This enables them to be easily compared against each other. The error probabilities are doubled as if only the positive half of the distribution existed, although this does not affect the comparison.



(a) (b) (c)

Figure 3: Analysis of algorithm on synthetic data; (a) using a Normal model the effects of perturbing the histogram are shown, (b) the thresholds selected under different models (c) the error rate under different models

Figure 3a shows the results of thresholding synthetic data distributed according to just the Normal model (plotted as the central continuous curve). Both axes are in units of σ. The graph illustrates the effect on the detected threshold value when the far end of the fitted straight line $(x_f, y_f) \equiv (x_f, 0)$ is shifted. When the end

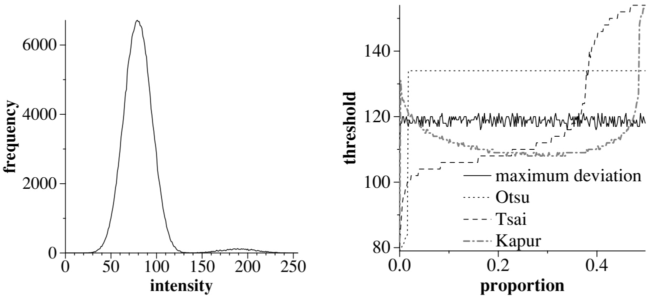
[1] If they are known to have higher values then the histogram can obviously be reflected before threshold analysis.

is very close to the peak the threshold changes rapidly, but only over a small range ($2\sigma - 3\sigma$). Otherwise if there is a wider separation between the start and end of the fitted line the selected threshold is stable. Although there is a consistent bias to increase the threshold value as the separation increases, the change is fairly insignificant. It should be added that, according to the statistics of the Normal, thresholding around the 3σ level is generally considered appropriate. The additional curves in the graph demonstrate the robustness of the approach. Those marked start were produced by perturbing the graph so that the start value y_s was rescaled by 0.75 or 1.25. The curve marked end (which lies very close to the uppermost curve) had the end value y_f set to $0.2 \times y_s$ (directly rescaling y_f would mostly have negligible effect). It can be seen in all cases that perturbing the graph has minimal effect on the selected threshold. Thus, as long the end of the fitted line is not too close to the peak then the method works well.

In figure 3b the results of thresholding data from the three model distributions are given. All are unstable when there is little separation between the ends of the fitted line. Increasing the power of the exponent produces a more stable result. There is little bias according to the $Q(t)$ model. Referring to figure 2 this is to be expected since the corner in the $Q(t)$ curve is well defined whereas the location of the $E(t)$ corner is much fuzzier.

Plotting the error functions (figure 3c) shows that the algorithm does not work well under the exponential model. The extreme peak and the poorly defined corner lead to high error rates (about 10%). In contrast, the other two distribution models lead to good results, only breaking down when the end of the line is very close to the peak ($< \sigma$). Thus our analysis predicts that the algorithm should work well unless the histogram is sharply peaked.

The proposed thresholding algorithm is now compared with three others from the literature, all of which are extremely disimilar in nature. Otsu's [11] is an example of the statistical-based approach; Tsai's [17] is based on the method of moments; and Kapur *et al.*'s [9] maximises the entropy of the partitioned histogram. A series of 512 × 512 images were synthetically generated containing pixels from two unevenly sized Normal distributions with mean intensities 80 and 190 and standard deviation 15. An example histogram is shown in figure 4a. Figure 4b shows how the thresholds selected by the algorithms vary as the proportion of the smaller class to the total image size varies. Not unexpectedly most have difficulties when the secondary class is very small. Otsu's method jumps up to a reasonable level and stablises there when the small peak's size is about 2% of the total image size. Tsai's method gradually creeps up to a reasonable level only when the peak size is about 35%. Kapur *et al.*'s method shoots up to reasonable levels almost immediately, although it does drift down to marginally underthreshold at the image when the small peak size is around $10\% - 40\%$. The proposed method is consistently around the 120 level, irrespective of the peak size. Since it is based on the shape of the histogram rather than the image as a whole it is more prone to the variations in the images due to noise. However, it can be seen that the variations in selected thresholds are not high.



(a)

(b)

Figure 4: Experiments with synthetic image data; (a) bimodal histogram with ratio of class sizes 1 : 50, (b) thresholds selected by various methods as a function of the relative class sizes

3 Examples

Having demonstrated the potential strengths of the proposed algorithm on synthetic data we now show its application to a variety of thresholding tasks, this time involving real data.

3.1 Edges

Figure 5a shows the Lena image, and the result of applying the Canny edge detector (inverted for display purposes) is given in figure 5b. Due to the non-maximal suppression stage filtering out secondary edges the histogram contains an enormous peak at zero. We eliminated this peak (figure 5c) and modified all programs to reflect this. The results show that simple thresholding (no hysteresis was applied) was best performed by the proposed method.

3.2 Change Detection

The second example is change detection, in which we have previously applied a related version of the thresholding method as well as several other new algorithms [15]. Not only is the image sequence poorly lit (both the original and difference image (figures 6a-b) have been contrast stretched) but the moving objects are extremely small (several birds on the grass in the

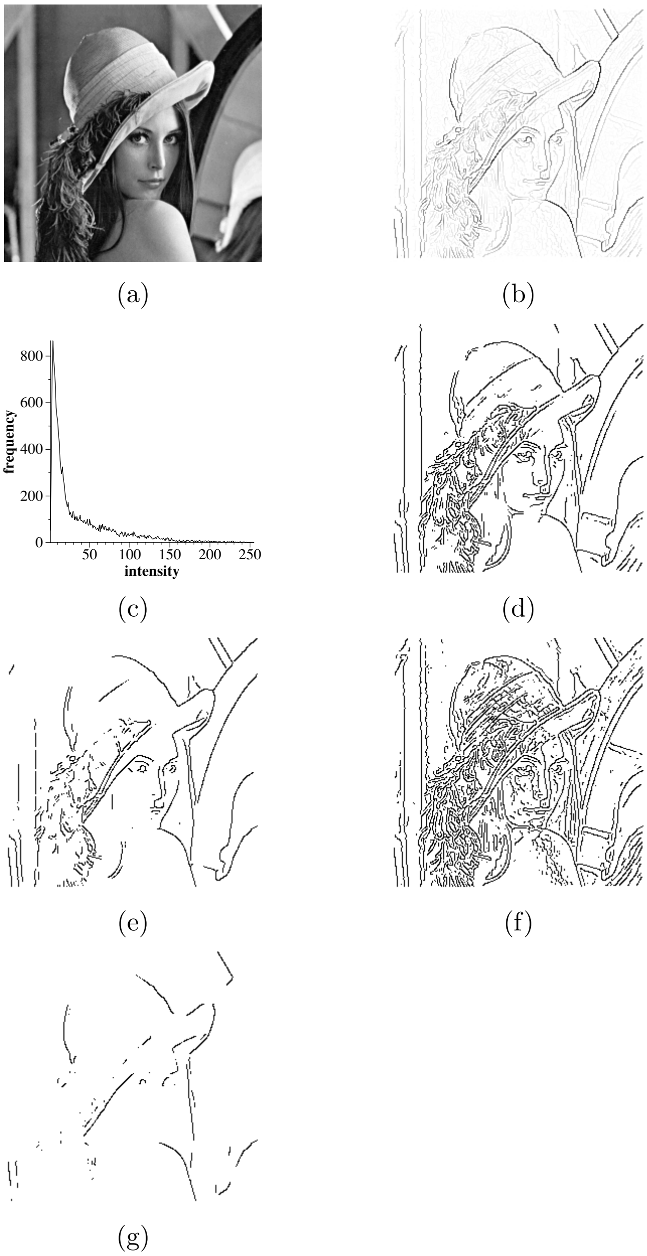


Figure 5: (a) Lena image, (b) inverted edge magnitudes, (c) edge histogram from 1-255, (d) proposed thresholding, (e) Otsu's thresholding, (f) Tsai's thresholding, (g) Kapur *et al.*'s thresholding

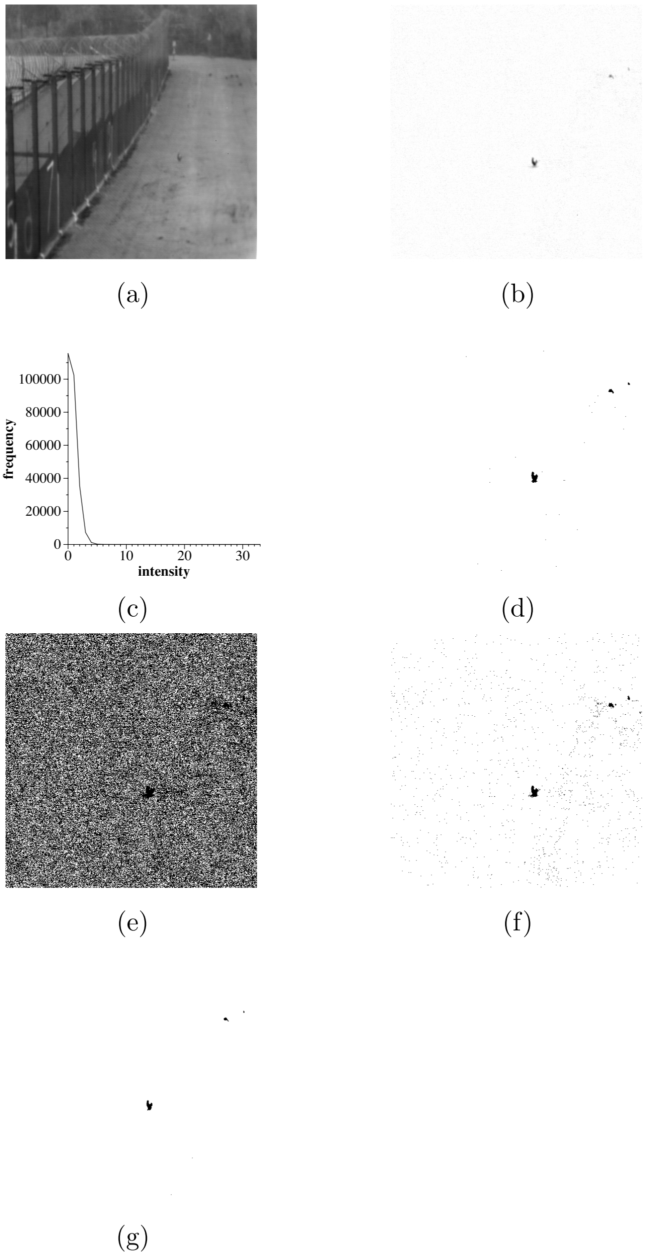
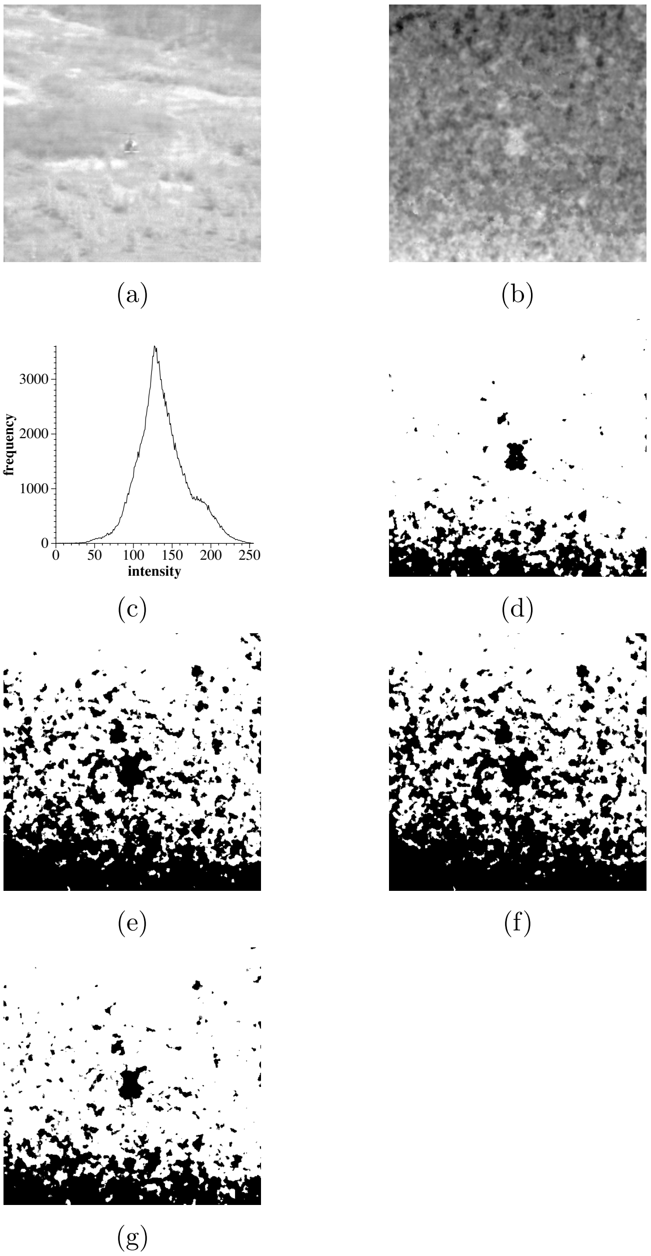


Figure 6: (a) SRDB018 first frame, (b) inverted difference image, (c) difference histogram, (d) proposed thresholding, (e) Otsu's thresholding, (f) Tsai's thresholding, (g) Kapur *et al.*'s thresholding

middle and far distance). This makes the difference histogram look perfectly unimodal (figure 6c). Nevertheless both the proposed method and Kapur *et al.*'s method perform well. Tsai's result is rather noisy while Otsu's breaks down altogether.

3.3 Optical Flow

Related to the example above, change detection is carried out in this case using the magnitude of the optical flow calculated by Anandan's algorithm [1]. This time since the helicopter in the image centre is being tracked (figure 7a) most of the image changes over the sequence, and so the the flow magnitude image is inverted before thresholding (figure 7b). It is clear that there cannot be perfect discrimination between the helicopter and the foreground. However, considering this, the result by the proposed thresholding method is good. Kapur *et al.*'s result is slightly inferior while Otsu and Tsai's results are markedly so.



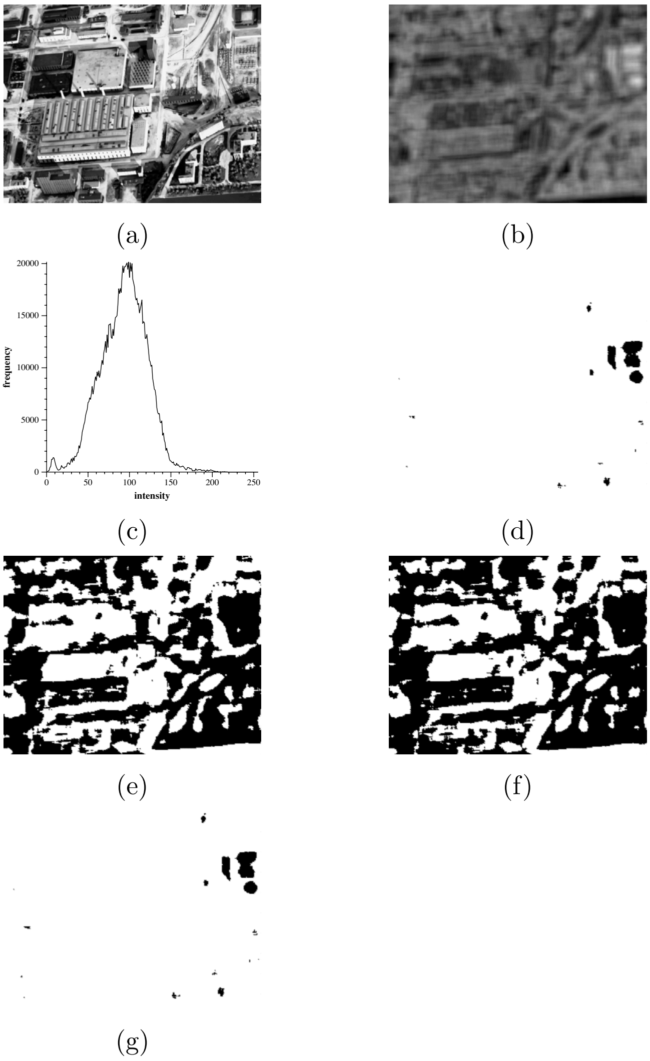
(a) (b) (c) (d) (e) (f) (g)

Figure 7: (a) chopper frame 6, (b) inverted magnitude optic flow, (c) flow magnitude histogram, (d) proposed thresholding, (e) Otsu's thresholding, (f) Tsai's thresholding, (g) Kapur *et al.*'s thresholding

3.4 Texture

As well as small regions of difference arising from change they can also arise from feature detection. This example creates a texture model, corresponding to an object or region of interest, and then calculates texture differences over the image relative to the model.

The image in figure 8a shows an aerial view of the RADIUS modelboard. A texture spectrum model [18] was trained from a small region containing parked cars. The differences between this and local texture spectra is shown (after inversion) in figure 8b. The histogram (figure 8c) contains a large peak of middling distance values, with tails giving relatively few small and large distances. We wish to detect the large distances (i.e. close matches since the difference image was inverted) to find other instances of cars in the image. Both the proposed algorithm and Kapur *et al.*'s algorithm work well, while both Otsu's and Tsai's thresholding totally breaks down (in the context of the task).



(a) (b) (c) (d) (e) (f) (g)

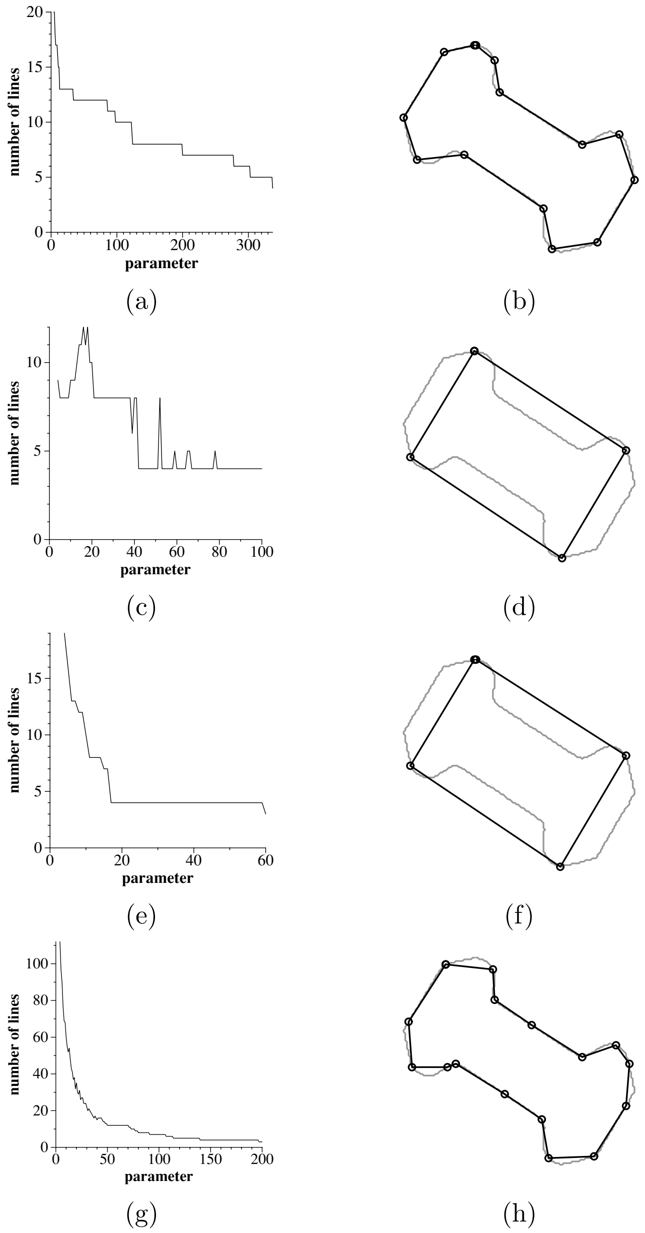
Figure 8: (a) RADIUS K3 image, (b) inverted texture distance, (c) histogram, (d) proposed thresholding, (e) Otsu's thresholding, (f) Tsai's thresholding, (g) Kapur *et al.*'s thresholding

3.5 Polygonal Approximation

The application of the thresholding method has already been successfully applied by Hill and Taylor [7] to the polygonal approximation problem. Rather than thresholding an image based on its histogram they used it to select an appropriate critical level for Zhu and Chirlian's [20] dominant point detection algorithm, analysing the histogram of the number of detected points indexed by the critical level.

We demonstrate further application of the approach to different polygonal approximation algorithms by Douglas and Peucker [5], Melen and Ozanian [10], Ramer [12], and Zhang *et al.* [19]. Figure 9 shows the histograms of the number of lines indexed by the algorithms' selected input parameters; these are distance threshold, filter size, distance threshold, and window size respectively. The algorithms are run until the resulting segmentation contains three lines, the minimum reasonable segmentation for a closed curve. Since the shapes of the histograms do not always follow the thresholding algorithm's expectations some of the resulting polygonal approximations are poor. The Douglas and Peucker histogram has a very fat tail, although the result is still good. The Melen and Ozanian histogram is very noisy, and given such discontinuities it is to be expected that the parameter selection process is unreliable. Although the Ramer histogram is well shaped the best defined corner in the histogram incorrectly sets the algorithm's distance parameter. Finally, the Zhang *et al.* histogram is also well shaped, and the result is reasonable although slightly oversegmented.

Since it is rather harder to judge the quality of the results of polygonal approximation compared to the other thresholding tasks several additional examples are given for the two better suited algorithms (see figure 10). The results are fairly consistent; the threshold algorithm selects good but rough approximations from the Douglas and Peucker algorithm and more detailed segmentations from the Zhang *et al.* algorithm.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

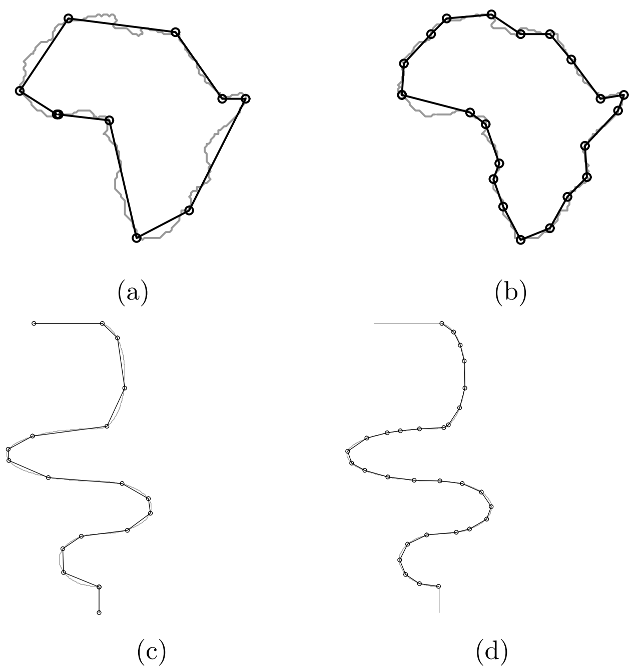
Figure 9: Plots of number of approximating lines against input parameter for various algorithms, and the approximation selected by the proposed algorithm; (a) and (b) Douglas and Peucker, (c) and (d) Melen and Ozanian, (e) and (f) Ramer, (g) and (h) Zhang *et al.*

3.6 Image Segmentation

In a similar manner parameter selection can be performed for image segmentation algorithms. In this example the split and merge [8] algorithm is applied to the angiogram image in figure 11a, and histograms are be formed by running through a range of the input parameters: the split and merge thresholds which in this instance are set equal to each other. Figure 11b shows the RMS histogram calculated as the differences between the original and segmented image (reconstructed using mean region intensities). Whereas this does not appear suitable for analysis using our method, the histogram of number of regions has the correct shape (figure 11c). The selected segmentation (reconstruction and boundaries) is shown in figure 11f-g, and looks reasonable (given the limitations of the split and merge algorithm). For comparison with that output, the results are also given for the input parameter divided and multiplied by a factor of three (figure 11d-e and figure 11h-i). The segmentations appear over- and under-segmented, indicating that the proposed algorithm at the very least approximately located the correct parameter value.

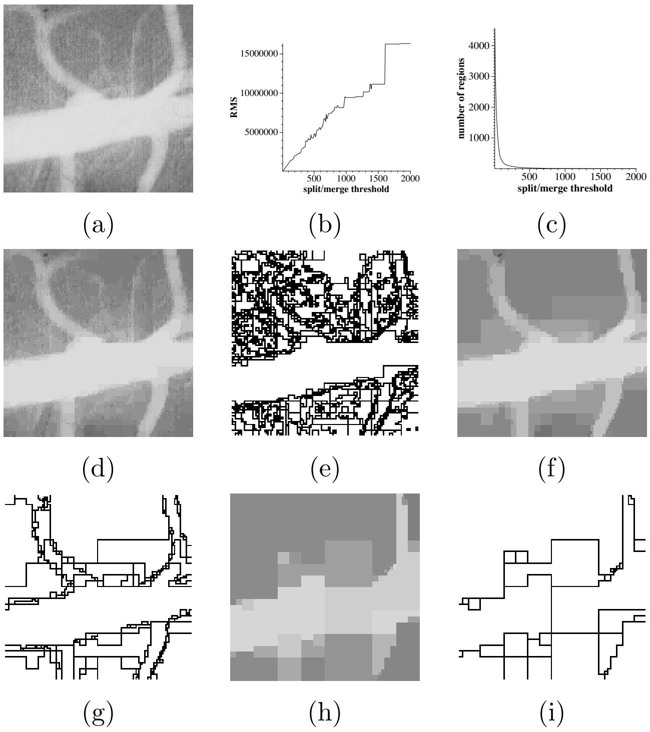
4 Discussion

A simple thresholding algorithm has been described. Unlike the majority of thresholding algorithms it is suitable primarily for essentially unimodal distributions.



(a) (b) (c) (d)

Figure 10: Polygonal approximations obtained using algorithms: (a) and (c) Douglas and Peucker; (b) and (d) Zhang *et al.*



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 11: Split and merge segmentation of 128 × 128 angiogram; (a) original image, (b) histogram of RMS error versus threshold (c) histogram of number of regions error versus threshold (d) and (e) segmented image at threshold value 50, (f) and (g) segmented image at selected threshold value 150, (h) and (i) segmented image at threshold value 450

Tests on synthetic data showed the feasibility of the approach. As long as 1) the mode is not so broad as to fill most of the histogram, and 2) the mode is not too strongly peaked (e.g. it is not exponential) then the mode is likely to contain a corner at its base that can be detected and is also a suitable threshold. Further tests were carried out on real data to select thresholds or other input parameters for various common tasks such as edge detection, change detection, texture detection, and the segmentation of curves and images. These demonstrated that the proposed technique worked well, often better than some standard thresholding algorithms that were run for comparison.

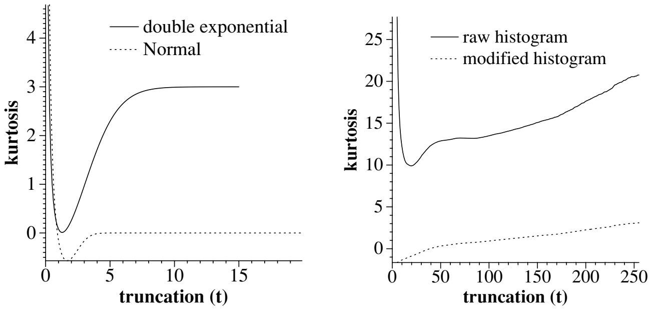
A possible remedy to the errors caused by strongly peaked histograms is to prescreen them so as to detect them, and signal their unsuitability for processing by the proposed algorithm. The degree of peakiness can be measured by the kurtosis $\kappa = \frac{\mu_4}{\sigma^2} - 3$ of the distribution; $\kappa > 0$ indicates greater peakiness than a Normal. In cases such as the edge detection example, where the approximate exponential (before removing bin 0) is one-sided, the histogram should be duplicated and reflected about the Y-axis to obtain a symmetric double exponential distribution. To avoid discrepancies arising from other entries in the histogram (e.g. a fat tail or secondary peak) the histogram could be truncated. Truncating at t (in units of σ) gives the following expressions for the kurtosis of the double exponential distribution

$$\frac{e^{\sqrt{2}t}\left(-6+6e^{\sqrt{2}t}-6\sqrt{2}t-6t^2-2\sqrt{2}t^3-t^4\right)}{\left(1-e^{\sqrt{2}t}+\sqrt{2}t+t^2\right)^2}-3$$

and the Normal distribution

$$\frac{e^{\frac{t^2}{2}}\left(-\left(\sqrt{2}t\left(3+t^2\right)\right)+3e^{\frac{t^2}{2}}\sqrt{\pi}\,\mathrm{erf}(\frac{t}{\sqrt{2}})\right)}{\sqrt{\pi}\left(\sqrt{\frac{2}{\pi}}t-e^{\frac{t^2}{2}}\,\mathrm{erf}(\frac{t}{\sqrt{2}})\right)^2}-3.$$

When they are plotted out (figure 12a) it is seen that as long as the truncation is not too severe ($> 3\sigma$) the kurtosis values can be estimated sufficiently well to be able to identify excessive peakiness. Applying this to the histogram produced from the full edge map in figure 5b (i.e. the raw histogram including bin 0) gives consistently high values of kurtosis for a range of truncation values (see figure 12b). If bin 0 is deleted and the histogram shifted down one ($H'_i = H_{i+1}$) then the measured kurtosis values of the modified histogram are much closer to those expected for a Normal distribution. The truncation point can be selected as 3σ with σ estimated from the complete histogram. From figure 5b we get $\sigma = 20.8$ which gives $\kappa = 13.2$, clearly indicating an excessively sharp peak.



(a) (b)

Figure 12: Kurtosis of truncated distributions; (a) the double exponential distribution, (b) the edge magnitude histogram of lena

A second area of investigation is the properties being histogrammed. So far we have mainly demonstrated image intensities (although these have been derived from different sources according to the various applications). However, as the polygonal approximation example demonstrated, other properties can also be histogrammed although not all are suitable. In that case RMS values were considered, and found not to be useful for the thresholding algorithm. Previously we have performed thresholding for change detection in which it was found that the image's Euler number as a function of threshold was a suitable input to the thresholding algorithm [15]. Many other such properties could be investigated as potentially useful for enabling thresholding of images where the intensity histogram is unsuitable. Some examples are the spatial randomness of the thresholded image (compare Rosin [15]), the average separation between thresholded regions, and shape properties of the regions such as average size, compactness, etc.

References

[1] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *Int. J. Computer Vision*, 2(3):283–310, 1989.

[2] A.D. Brink and N.E. Pendock. Minimum cross-entropy threshold selection. *Patt. Recog.*, 29(1):179–188, 1996.

[3] F.H.Y. Chan, F.K. Lam, and H. Zhu. Adaptive thresholding by variational method. *IEEE Trans. IP*, 7(3):468–473, 1998.

[4] S. Dizenzo, L. Cinque, and S. Levialdi. Image thresholding using fuzzy entropies. *IEEE Trans. on SMC B*, 28(1):15–23, 1998.

[5] D.H. Douglas and T.K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer*, 10:111–122, 1973.

[6] J.A. Gong, L.Y. Li, and W.N. Chen. Fast recursive algorithms for 2-dimensional thresholding. *Patt. Recog.*, 31(3):295–300, 1998.

[7] A. Hill and C.J. Taylor. A method of non-rigid correspondence for automatic landmark identification. In *Proc. British Machine Vision Conf.*, pages 323–332, 1996.

[8] S.L. Horowitz and T. Pavlidis. Picture segmentation by a tree traversal algorithm. *J. ACM*, 23(2):368–388, 1976.

[9] 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. *CVGIP*, 29(3):273–285, 1985.

[10] T. Melen and T. Ozanian. A fast algorithm for dominant point detection on chain-coded contours. In *Proc. CAIP*, 1993.

[11] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. SMC*, 9:62–66, 1979.

[12] U. Ramer. An iterative procedure for the polygonal approximation of plane curves. *CGIP*, 1:244–256, 1972.

[13] A. Rosenfeld and P. De La Torre. Histogram concavity analysis as an aid in threshold selection. *IEEE Trans. SMC*, 13:231–235, 1983.

[14] P.L. Rosin. Edges: saliency measures and automatic thresholding. *Machine Vision and Applications*, 9(4):139–159, 1997.

[15] P.L. Rosin. Thresholding for change detection. In *Proc. ICCV*, pages 274–279, 1998.

[16] P.K. Sahoo, S. Soltani, A.K.C. Wong, and Y.C. Chen. A survey of thresholding techniques. *CVGIP*, 41:233–260, 1988.

[17] W.H. Tsai. Moment-preserving thresholding. *CVGIP*, 29:377–393, 1985.

[18] L. Wang and D.C. He. Texture classification using texture spectrum. *Patt. Recog.*, 23:905–910, 1990.

[19] X. Zhang, R.M. Haralick, and V. Ramesh. Corner detection using the MAP technique. In *Proc. ICPR*, pages 549–551, 1994.

[20] P. Zhu and P.M. Chirlian. On critical-point detection of digital shapes. *IEEE Trans. PAMI*, 17(8):737–748, 1995.