Thresholding Using an Illumination Model

J. R. Parker
Laboratory for Computer Vision
University of Calgary
Calgary, Alberta, Canada

ABSTRACT

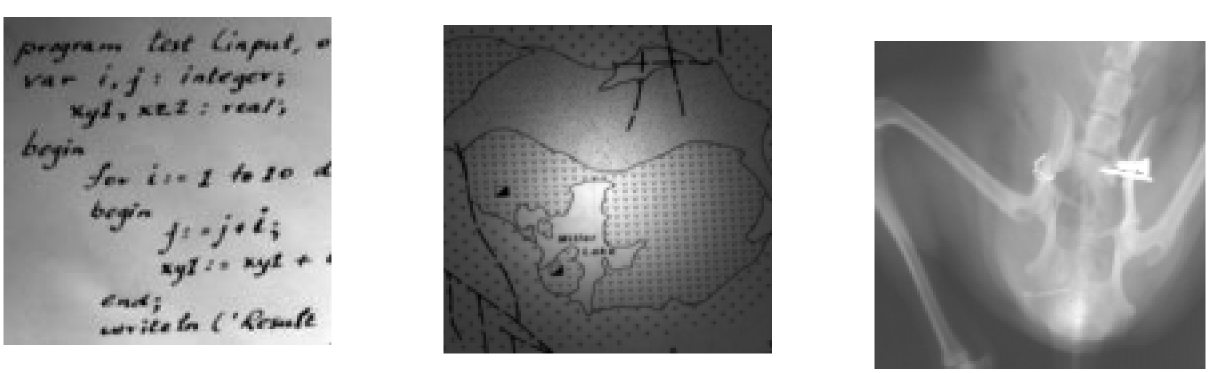
Most grey level thresholding methods produce good results in situations where the illumination gradient in the original raster image is regular and not too large. In other cases, such as a large linear change in illumination, a satisfactory bi-level image cannot be produced. If the object pixels can be identified in a variety of positions throughout the image, these can be used to construct a surface whose height is related to illumination at each pixel. This estimate can be used to produce a threshold for each pixel. The method described here uses the Shen-Castan edge detector to identify object pixels, and creates a surface using a moving least squares method that can be used to threshold the image.

I. Introduction

The purpose of grey level thresholding is to extract from an image those pixels which represent an object. The object is often text or other line image data (graphs, maps). After thresholding the image, the object pixels have all one grey level and the background pixels have another. The best threshold is the one that selects all the object pixels and maps them to black. Unfortunately, it is not possible in general to find a single threshold that is ‘best’ for an arbitrary grey-level image[4, 6], and it is a simple matter to construct an image that cannot be successfully thresholded with a single value[10]. This type of situation can also arise in real images because of noise or non-uniform lighting.

If a single threshold is not acceptable, then the option that remains is to use a number of thresholds, each over a small sub-region of the image. The maximum number of subregions is the same as the number of pixels in the image; moreover, any larger region may be subject to distortion through illumination effects, although smaller distortions than would be seen over the entire image. A starting point for devising a thresholding method can be to select a different threshold for each pixel. The method will decide what these threshold values are to be based on *local* properties of the image. Our confidence in the local threshold decreases with the distance from a known object pixel. Figure 1 shows three images, each having an illumination problem that would make them difficult to threshold properly.

Our approach to thresholding is based on the principle that objects in an image provide the high spatial frequency component and illumination consists mainly of lower spatial frequencies. These two were multiplied together to produce the image. Another way to look at this is to say that the objects in an image will produce small regions with a relatively large intensity gradient, those being at the boundaries of objects, whereas other areas ought to have a relatively small gradient[5,6]; this fact is used in many edge enhancement algorithms. In this way a sample of the object pixels in an image can be found by looking for regions of high gradient and assum-



ing that these pixels belong to an object that would appear as distinct in a thresholded picture[1].

The thresholding method that is being proposed involves first locating ‘objects’ in an image by using the Shen-Castan edge detector[8] to locate pixels that belong to boundaries of objects. This edge detector has good localization properties, and a pixel that has been determined to be on an edge will be assumed to be a part of an object. The grey levels at edge pixels should be indicative of the levels elsewhere in object regions. A surface is produced that fits the levels at the edges, and this surface is presumed to give reasonable estimates of likely grey levels at object pixels that do not lie on an edge. Pixels significantly above this surface will be assumed to belong to the background, and those at or below it will belong to the object. This method is capable of thresholding images that have been produced in the context of variable illumination. This algorithm is called Edge Level Thresholding (*ELT).*

2. Shen-Castan (ISEF) Edge Detection

There have been numerous efforts to define ‘optimality’ in the context of edge detection, and edge detection algorithms have been based on these definitions in a one-to-one mapping. A well-known example of this sort of work is the Canny edge detector[7] which optimizes a combination of the signal to noise ratio, the localization of the edges, and enforces a single response to a single edge. Once the definition of optimal has been determined it is used to derive a filter that implements it.

Shen and Castan[8] suggested a different optimality criterion, using a mono-step edge model. The edge detection filter would have a maximal response at the exact edge position, and would minimize the noise energy in both the smoothed output and the differential input. While the Canny filter is an approximation to the optimal, the ISEF is precise, and has better signal to noise ratios.

3. Fitting A Surface To The Edges: Moving Least Squares

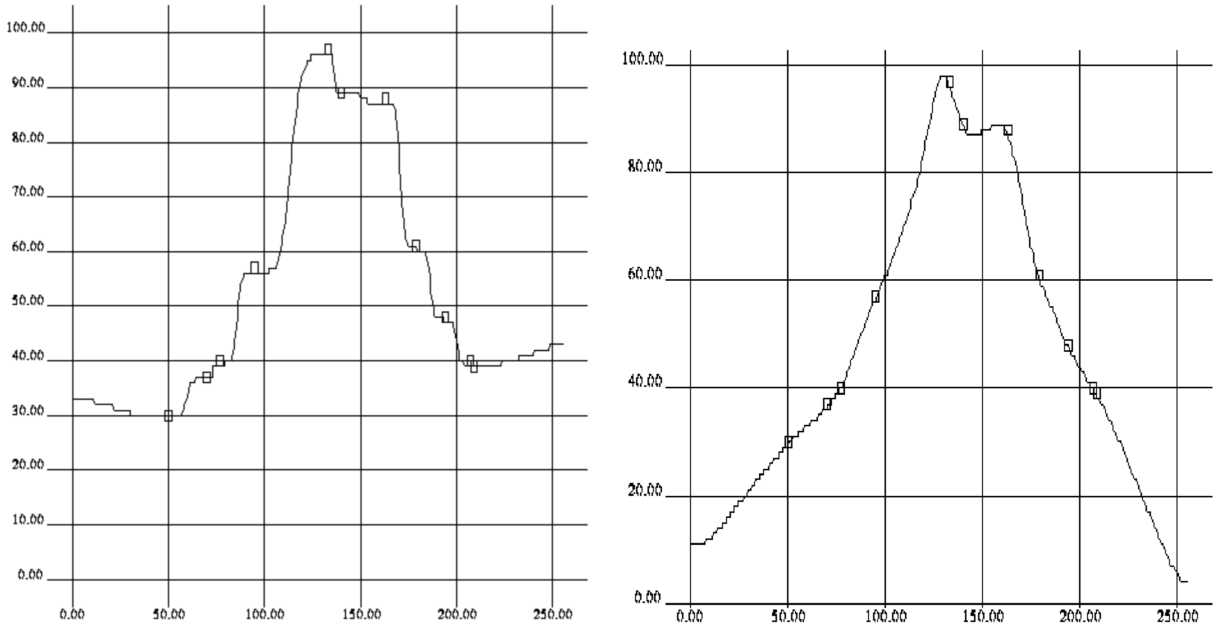
The method we use to fit a surface to the edge points is a moving least-squares (MLS) scheme[9]. This involves solving a weighted least-squares problem at each point in the plane. That is, if where N is the number of data points which are given

$$J(x, y) = \sum_{i=1}^{N} w_i(x,y)(I(x_i,y_i) - S(x_i,y_i))$$

by $I(x_i,y_i)$, $S(x, y) = ax + by + c$, and $w_i(x, y)$ are weights, then we find values for and **c** so that J(x,y) is minimized at each point (x,y) in the plane. The weights depend on the evaluation points, and hence the requirement that we perform this minimization at each point.

The weight function $w_i(x, y)$ has several important properties.It essentially weights the data point (x_i, y_i) inversely according to its distance from the current evaluation point (x,y). If the data is further than some specified distance **h** from (x,y), we assume that it should have no bearing whatsoever on the height of the surface at that point, and so the weight is zero. Another parameter for the weight function, **d**, determines the fidelity of the surface to the data. When **d** is zero the weight is essentially infinite when the evaluation point is also a data point - the result is a surface that actually passes through all of the data, and this can lead to extreme fluctuations. As **d** increases towards 1 the fidelity increases and the surface relaxes, tending to average out fluctuations. When **d=1** our weight function is defined to be constant for all data, and no longer has compact support. The resulting surface is simply the standard least-squares planar approximation to the entire data set.

Since at every point we are looking for a least-squares plane, there is a rigid mathematical requirement that we have at least 3 data values in the disk of radius **h** centered at each point in the image. Without these, the linear system will be underdetermined, and we won’t be able to find a solution to the least-squares problem.



(a) (b)

Figure 2 - a) A weighted average applied to a set of data. b) MLS applied to the same data set.

Others working on this problem[13] have suggested methods for getting an approximation to the edge data, but these are all interpolants, and as we have previously mentioned, this is not necessarily desirable. One method described is a moving weighted average, which is simply an MLS method with $S(x, y) = a$. The resulting surface will have horizontal tangent planes or ‘flat-spots’ at each of the data points[12].

In addition, as the evaluation moves away from data, the value of the weighted average will tend to the actual mean of the data. This may cause unusual artifacts if the illumination gradient is actually linear. Figure 2a shows an example of this method restricted to the one-dimensional case. Figure 2b shows an example of our method applied to the same data.

4. Implementation and Results

The ELT thresholding software consists of three major modules: the ISEF edge detector, the MLS surface fit, and the thresholding module. The function and sequence of execution is best described by using an example. Consider the image of Figure 1b, in which a bright Gaussian spot has been superimposed on the image of a map. The first step is edge detection by ISEF, and the result is shown in Figure 3a. Notice that clean edges are found even in the dark areas of the image. This is the secret of the ELT method: ISEF finds edges *very* well, and these edges are well localized.

Next, the grey levels at all edge pixels are used to form the basis of a surface, and the levels at the non-edge pixels are estimated from this surface, as found by the MLS procedure. For this case the surface is shown in Figure 3b as grey levels and in Figure 3c as a wire frame graphic. The value of the function at the edge pixels will be very near to the actual value of the corresponding image pixel, and will be assumed to be near to the value of non-edge object pixels as well. The final stage is a pass through the image, setting all pixels to zero if they are less than the value of the fit function +10, and setting them to MAX (255) otherwise. The results can be seen in Figures 3d-f.

In general, the results of ELT thresholding are better than other algorithms in situations of poor illumination, especially when compared to single-threshold methods. Standard methods give results that have large black areas where the illumination reaches low levels, and the objects can’t be determined from the background. ELT permits widely varying thresholds across the image.

5. Conclusions

The ELT algorithm has been compared with other thresholding methods, and none has given the same results in widely varying illumination environments. ELT is quite slow at this time, but could be optimized and made parallel. There are a few parameters to the algorithm that can be adjusted: for example, the MLS code we use

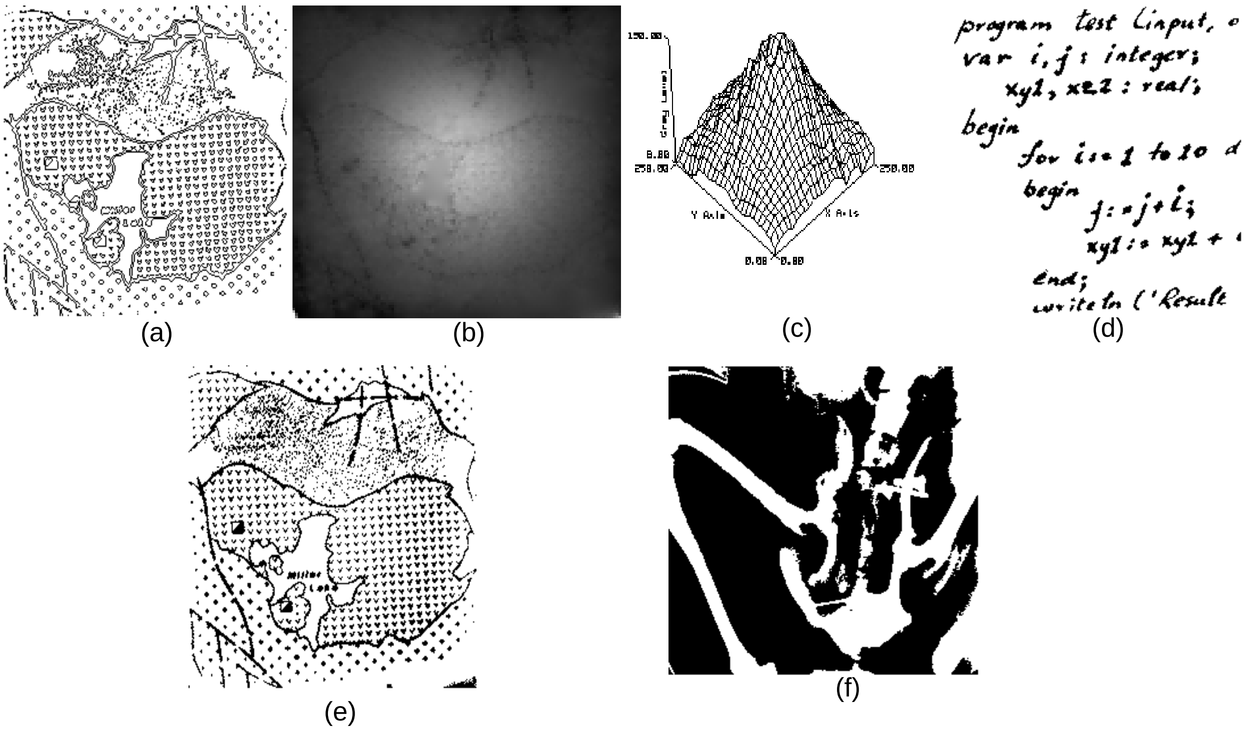


Figure 3 - a) Shen-Castan edges for Fig 1b. b) Surface fit to edge pixels. c) Surface drawn in 3-D. d) Thresholded version of Fig 1a. e) Thresholded version of Fig 1b. f) Thresholded version of Fig 1c.

relies on a fixed value for the radius **h** for the entire image. This presents some problems since in some regions a large radius is required so that we have at least 3 points, while in other areas the points are so dense that the same size disk will include hundreds of points.

6. Acknowledgments

This research has been supported by the Natural Sciences and Engineering Research Council of Canada.

7. References

[1] Bracho, R., and Sanderson, A.C.,"Segmentation of Images Based On Intensity Gradient Information", *Proc. IEEE Computer Soc. Conf. on Computer Vision and Pattern Recognition*, San Francisco, June 19-23, 1985. Pp. 341-347.

[2] Otsu, N., "A Threshold Selection Method From Grey-level Histograms", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 9 No. 1, 1979. Pp 377-393

[3] Perez, Arnulfo and Gonzalez, R.C., “An Iterative Thresholding Algorithm for Image Processing”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, No. 6, November 1987.

[4] Prewitt, j. M. S., “Object Enhancement and Extraction”, in *Picture Processing and Psychopictorics*, (Lipkin and Rosenfeld Eds.), Academic Press, New York. 1970.

[5] Stockham, T.G., “Image Processing in the Context of a Visual Model”, *Proceedings of the IEEE*, Vol. 60, 1972. Pp 828-842

[6] Wilson, Roland and Spann, Michael, “Image Segmentation and Uncertainty”, John Wiley & Sons Inc., 1988.

[7] Canny, J., “A Computational Approach To Edge Detection”, *IEEE-PAMI*, 8(6):679-698, 1986.

[8] Shen, J. and Castan, S., “An Optimal Linear Operator For Step Edge Detection”, *CVGIP: Graphical Models and Image Processing*, 54(2) Pp 112-133, 1992.

[9] Salkauskas, K. and Lancaster, P., “Curve and Surface Fitting, An Introduction”, Academic Press, 1981.

[10] Parker, J.R., “Grey Level Thresholding In Badly Illuminated Images”, *IEEE- PAMI*, Vol 13, No. 8, 1991.

[11] Cline, A. K. and Renka, R. L., “A Storage-efficient method for construction of a Thiessen triangulation”, *Rocky Mountain Journal of Mathematics*,14. 119-139 (1984).

[12] Lancaster P. and Salkauskas K., “Surfaces Generated by Moving Least Squares Methods”, *Math. Comp.* 37, Pp. 141-158,1981.

[13] Yanowitz, S.D. and Bruckstein, A.M.,”A New Method For Image Segmentation”, *Computer Vision, Graphics, and Image Processing*, 46, 1989. Pp. 82-95.