Gaussian-weighted moving-window robust automatic threshold selection

Wilkinson, Michael

Published in:
COMPUTER ANALYSIS OF IMAGES AND PATTERNS, PROCEEDINGS

Publication date:
2003

Gaussian-Weighted Moving-Window Robust Automatic Threshold Selection

Michael H. F. Wilkinson

Institute for Mathematics and Computing Science,
University of Groningen, P.O. Box 800, 9700 AV Groningen, The Netherlands,
michael@cs.rug.nl
http://www.cs.rug.nl/~michael/

Abstract. A multi-scale, moving-window method for local thresholding based on Robust Automatic Threshold Selection (RATS) is developed. Using a model for the noise response of the optimal edge detector in this context, the reliability of thresholds computed at different scales is determined. The threshold computed at the smallest scale at which the reliability is sufficient is used. The performance on 2-D images is evaluated on synthetic an natural images in the presence of varying background and noise. Results show the method deals better with these problems than earlier versions of RATS at most noise levels.

1 Introduction

In all applications of thresholding, correct selection of the threshold is the key issue, and many methods for automatic selection of optimal thresholds have been published[1–4, 8]. Ideally, thresholds should be computed locally, adapting to the local image statistics, to deal properly with a locally varying background, or variations in the grey level of objects, both of which may occur in a single image [8]. An example is shown in Fig. 1, in which a local variations in object intensity are compensated through local threshold selection.

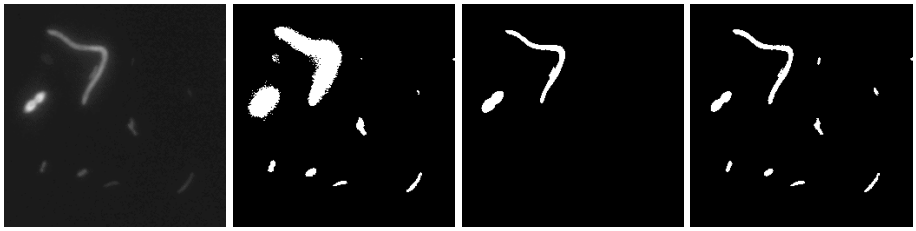
In this paper I will extend a local thresholding method, called Robust Automatic Threshold Selection (RATS) [1]. A new, moving-window version of the algorithm using Gaussian convolution will be developed. This versions will be extended to a multi-scale method, in which the smallest scale at which reliable thresholds can be computed is finally used. To do this, the effect of Gaussian noise on the computed threshold is derived. Finally, the method is evaluated first on synthetic images with varying noise levels, and later on natural ones.

The applications for the method include 2-D microscopic images of microorganisms, and 3-D angiograms.

2 Robust Automatic Threshold Selection

RATS [1] is a method for bilevel thresholding of grey scale images, which has been applied to images of bacteria [5, 6]. It is based on a simple image statistic,

N. Petkov and M.A. Westenberg (Eds.): CAIP 2003, LNCS 2756, pp. 369-376, 2003



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

Fig. 1. Local thresholding: (a) fluorescence image of bacteria; (b) global thresholding using original RATS algorithm without noise correction; (c) global thresholding using RATS algorithm with square Sobel gradient filter and noise correction; (d) locally thresholded result using RATS with the quad-tree approach from [5].

which is the average of grey levels weighted by the edge strength at each point. Kittler et al. [1] show that the optimal threshold in a noise-free image T is

$$T = \frac{\sum e(x,y)p(x,y)}{\sum e(x,y)}, \tag{1}$$

in which $p(x,y)$ is the grey level at (x,y) and the edge strength e is given by

$$e(x,y) = \max(|g_x(x,y)|, |g_y(x,y))|, \tag{2}$$

with

$$g_x(x,y) = p(x-1,y) - p(x+1,y) \quad \text{and} \quad g_y(x,y) = p(x,y-1) - p(x,y+1). \tag{3}$$

Initially the optimality of T was proved only for gradient operator in (2), and for straight edges. It has since been shown that any edge detector with an even response to a step edge at the origin will yield the same optimal result [7]. In particular, the gradient detector

$$g^2(x,y) = g_x^2(x,y) + g_y^2(x,y) \tag{4}$$

shows no curvature bias, is rotation invariant, and has reduced noise bias. However, the reduced noise bias comes at the expense of increased variance, which can be countered by using Sobel filter kernels to compute x and y derivatives [7]. The method readily extends to 3-D images, by replacing the edge detectors to their 3-D counterparts.

In the presence of noise T is biased towards the most common category in the image (usually background) [1]. This noise bias is counteracted by using a threshold on the edge strength below which the pixels receive zero weight. The statistic now becomes

$$T_\lambda = \frac{\sum w_\lambda(x,y)p(x,y)}{\sum w(x,y)}, \tag{5}$$

with

$$w_\lambda(x,y) = \begin{cases} g^2(x,y) & \text{if } g^2(x,y) > \lambda^2\eta^2 \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

in which η is the standard deviation of the image noise, and λ is an adjustable parameter, which depends on the actual edge strength used. For the edge strength defined in (2) it was shown empirically that $\lambda = 5$ is a good choice for Gaussian noise [7].

2.1 Local Application of RATS

RATS lends itself well to local application [1] for two reasons: (i) the statistic in (5) is robust against noise, and (ii) it is easy to check whether a region contains an edge by checking whether the denominator in (5) is above some threshold [1, 7]. Ideally, we want to compute the threshold in an isotropic surroundings of each pixel. This can be done using a moving-window version of RATS, which can be written as the ratio of two convolutions

$$T_h(x,y) = \frac{(\Pi_h * (w \cdot p))(x,y)}{(\Pi_h * w)(x,y)}, \tag{7}$$

in which $*$ denotes convolution and $\Pi_h(x,y)$ is given by

$$\Pi_h(x) = \begin{cases} 1 & \text{if } |x| \le h \text{ and } |y| \le h, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

One problem with (7) is that T_h is undefined for all pixels where $(\Pi_h * w)(x,y) = 0$. Thus wherever the distance between edges is greater than the width of the window, no threshold is computed. Besides, the square convolution kernels are not isotropic. However, the convolution formalism allows generalization of the algorithm to other convolution kernels, e.g. Gaussian. Gaussian kernels are separable, and have infinite impulse response (IIR), and so will contribute over the entire image. Besides, can be computed quickly using a recursive implementation which has an IIR [9]. We arrive at

$$T_\sigma(x,y) = \frac{(G_\sigma * (w \cdot p))(x,y)}{(G_\sigma * w)(x,y)}, \tag{9}$$

with G_σ a Gaussian with zero mean and standard deviation of σ. A further advantage (9) over (7) is that edges close to the current pixel are given higher weights than distant edge. However, despite their infinite impulse response, G_σ falls off so rapidly that at large distances from edges thresholds may become unreliable, because a few remaining nearby noise edges may outweigh distant true edges. To counter this, multi-scale approach can be used, by computing T_σ with $\sigma = \sigma_0, 2\sigma_0, 4\sigma_0, \ldots, \sigma_{\max}$, and using the lowest σ for which T_σ can be computed reliably.

2.2 Selecting the Correct Scale

To select the correct scale we need to understand how the noise influences the statistic T_σ. Let us assume that the noise in an image is Gaussian with zero mean and standard deviation η. Its distribution is simply

$$p(x)dx = \frac{1}{\eta\sqrt{2\pi}} e^{-x^2/2\eta^2} \tag{10}$$

This means that the probability distribution p_{g_x} of g_x (or g_y or g_z) is

$$p_{g_x}(x)dx = \frac{1}{2\eta\sqrt{\pi}} e^{-x^2/4\eta^2}. \tag{11}$$

The probability distribution p_1 of g_x^2 is

$$p_1(x)dx = p_{g_x}(\sqrt{x})d\sqrt{x} = \frac{1}{2\eta\sqrt{2\pi x}} e^{-x/4\eta^2}. \tag{12}$$

In 2-D, the probability distribution p_2 of g^2 in (4) is

$$p_2(x)dx = (p_1 * p_1)(x)dx = \frac{1}{4\eta^2} e^{-x/4\eta^2}, \tag{13}$$

The 3-D counterpart of g^2 has a probability distribution p_3 given by

$$p_3(x)dx = (p_2 * p_1)(x)dx = \frac{\sqrt{x}}{4\eta^3\sqrt{2\pi}} e^{-x/4\eta^2}. \tag{14}$$

To select the lowest scale at which the denominator in (9) becomes significantly different from the value expected from noise, we need both the mean value $\langle w_\lambda \rangle$ and the variance σ_w^2 of w_λ. Using (6) and (13) the distribution p_w of w_λ is

$$p_w(x) = \begin{cases} \delta(x) \int_0^{\lambda^2\eta^2} p_2(x')dx' & \text{if } x < \lambda^2\eta^2, \\ p_2(x) & \text{otherwise.} \end{cases} \tag{15}$$

It can be seen from (13) that the probability p_λ that $x > \lambda^2\eta^2$ is $e^{-\lambda^2/4}$. Therefore, the expected value $\langle w_\lambda \rangle$ is

$$\langle w_\lambda \rangle = \int_{\lambda^2\eta^2}^{\infty} \frac{x}{4\eta^2} e^{-x/4\eta^2} dx = e^{-\frac{\lambda^2}{4}} \left(1 + \frac{\lambda^2}{4}\right) 4\eta^2 \tag{16}$$

and the variance σ_w^2 is

$$\sigma_w^2 = e^{-\frac{\lambda^2}{4}} \left((1 - e^{-\frac{\lambda^2}{4}}) \left(1 + \frac{\lambda^2}{4}\right)^2 + 1 \right) 16\eta^4 \approx e^{-\frac{\lambda^2}{4}} \left(1 + \frac{\lambda^2}{4}\right)^2 16\eta^4 \tag{17}$$

in the 2-D case. Thus, the standard deviation σ_w is approximately

$$\sigma_w \approx e^{-\frac{\lambda^2}{8}} \left(1 + \frac{\lambda^2}{4}\right) 4\eta^2. \tag{18}$$

For $\lambda \geq 5$ the approximation is accurate to within 6%. The mean noise response in the Gaussian-convolved edge image is just $\langle w_\lambda \rangle$, whereas the variance of the noise response σ_{Gw}^2 is

$$\sigma_{Gw}^2 = \sigma_w^2 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} G_{\sigma_i}^2(x,y)dxdy = \frac{1}{4\pi\sigma_i^2}\sigma_w^2. \tag{19}$$

We can select the lowest scale σ_i for which the denominator of (9) is larger than a threshold $T_{Gw} = \langle w_\lambda \rangle + 3\sigma_{Gw}$, or

$$(G_{\sigma_i} * w)(x,y) \geq T_{Gw} = e^{-\frac{\lambda^2}{8}}\left(1 + \frac{\lambda^2}{4}\right)\left(e^{-\frac{\lambda^2}{8}} + \frac{3}{2\sigma_i\sqrt{\pi}}\right)4\eta^2 \tag{20}$$

Note that as $\sigma_i \to \infty$, the variance $\sigma_{Gw}^2 \to 0$, and so $T_{Gw} \to \langle w_\lambda \rangle$. When using the Sobel kernels in our initial edge detector, the only thing that needs to be changed in this calculation is to replace η with $\sqrt{5}\eta/4$ [7].

3 Algorithm

Let array p contain the original image, array w store the weights, array wp the product $w_\lambda p$, array T the threshold, and a boolean array q the binary output image. All arrays are of the same size as the original image. Let the value Inv denote an invalid threshold (e.g., some value $> \max_{x,y}(p(x,y))$). Finally, we have T_λ to store the global threshold according to (5).

The multi-scale, Gaussian-weighted, moving-window RATS algorithm is summarized in Fig. 2. The only input the algorithm needs is the original image, the desired value of λ, and the image noise η. After computing of w_λ and storing in in w, we compute the product image $w_\lambda p$, and the global threshold T_λ. We initialize all elements of T to Inv, and convolve both w and wp with G_{σ_0} at the lowest scale. After this initial phase, we loop through all scales but the last. During each loop we first compute T_{Gw} at that scale, and then compute T_{σ_i} for all pixels (x,y) which have not yet been assigned a valid threshold (i.e., $T(x,y) = Inv$), *and* for which $w(x,y) \geq T_{Gw}$. We then compute $G_{\sigma_{i+1}} * w$ from $G_{\sigma_i} * w$ by convolving $G_{\sigma_i} * w$ with $G_{\sqrt{3}\sigma_i}$, and likewise for $G_{\sigma_{i+1}} * wp$. At the largest scale, a similar operation is performed, but here $T(x,y)$ is set to the global threshold T_λ if no threshold can be computed at that scale. Finally, for each pixel $q(x,y)$ is set to *true* if $p(x,y) > T(x,y)$ and to false otherwise.

4 Results

The algorithm was implemented, using the Sobel gradient, $\lambda = 7$, and four scales ranging from $\sigma = 2$ to $\sigma = 16$, and tested on synthetic 2-D images of 256×256 pixels containing objects of different sizes and different or constant contrast with respect to the local background. The constant object-intensity images (see Fig. 3(b)) served as reference segmentation for themselves and the corresponding variable object-intensity images as in Fig. 3(a).A background slope running

```
1.  For all pixels (x, y) compute w_λ(x, y) from input image p and store in w.
2.  Compute product image w_λ p and store in wp.
3.  Compute T_λ from wp and w.
4.  Set all values in T to Inv.
5.  Convolve w and wp with G_{σ_0} and store in w and wp respectively.
6.  For all scales i = 0, 1, ..., max −1 do
7.      Compute T_{Gw} for this scale
8.      For all pixels (x, y) with T(x, y) ≠ Inv
9.          if w(x, y) ≥ T_{Gw} then
10.             T(x, y) ← wp(x, y)/w(x, y).
11.     Convolve w and wp with G_{√3 σ_i} and store in w and wp respectively.
12. Compute T_{Gw} for σ_max
13. For all pixels (x, y) with T(x, y) ≠ Inv
14.     if w(x, y) ≥ T_{Gw} then
15.         T(x, y) ← wp(x, y)/w(x, y).
16.     else
17.         T(x, y) ← T_λ
18. For all pixels (x, y)
19.     q(x, y) ← p(x, y) > T(x, y)
```

Fig. 2. The multi-scale, Gaussian-weighted, moving-window RATS algorithm. Note that $\leftarrow$ denotes assignment.

from 0 at the left and height h_s at the right, and Gaussian noise with standard deviation η were added. The fraction of misclassified pixels was computed as a function of η and h_s. The results are shown in Fig. 3(c)-(h). Using all four scales the performance is generally good up to an η of about 8 (corresponding to an S/N-ratio of about 8 for the faintest objects) for objects of varying intensity (Fig. 3(c), (d) and (f)). Fig. 3(d) shows how omission of the lowest scale results in segmentation errors where faint objects lie close to bright. Fig. 3(f) shows a comparison of the new method with global thresholding by Otsu's method [2]. The latter performs badly at all noise levels, because it classifies fainter foreground objects as background. For the former method the error fraction rises sharply beyond $\eta = 8$ to about 7% at $\eta = 32$. An example is shown in Fig. 3(e). The slope h_s has no impact on segmentation quality (not shown). Segmentation of Fig. 3(b) is easier, the errors never exceeding 0.5 % (not shown). Using just a single scale σ results in rather poor segmentation, unless the smallest scales are used Fig. 3(g). Apparently, the lowest values of σ have a high impact on the quality of segmentation. Finally, Fig 3(h) shows a comparison between the earlier quad-tree method and the new algorithm for $h_s = 16$. Only at $\eta = 16$ does the old method perform better, suggesting that the way noise is dealt with using (20) could be improved, perhaps by giving less weight to the lowest scale. The results in Fig. 3(g) also suggest that using $\sigma = 4$ at $\eta = 16$ is better than the multi-scale approach.

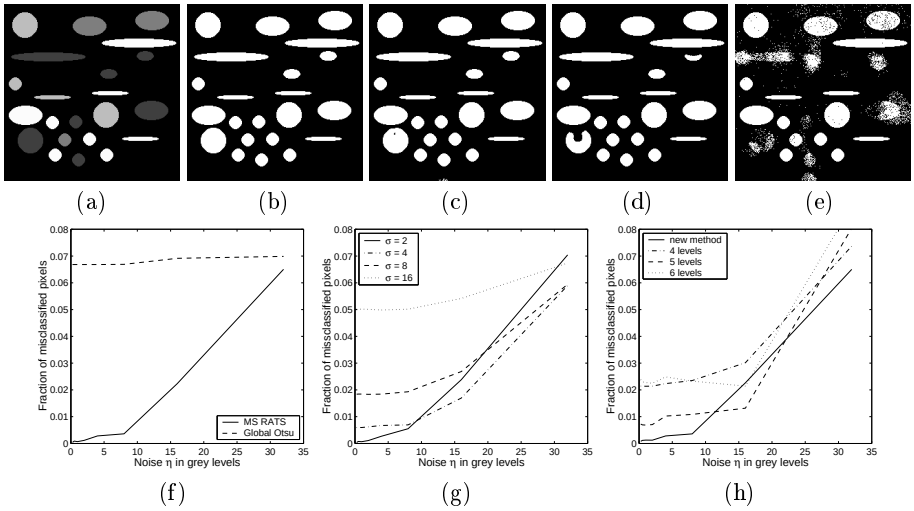
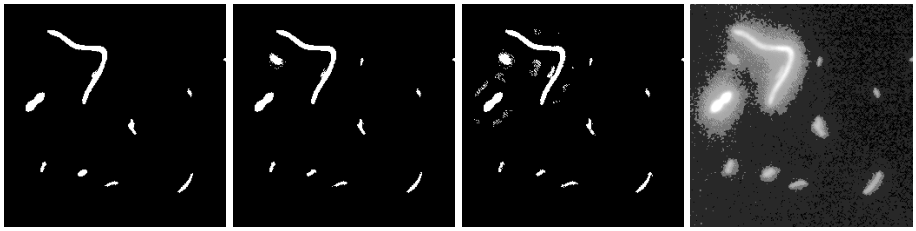


Fig. 3. Segmentation of synthetic images of ellipses: (a) synthetic image of ellipses of differing intensities; (b) same as (a) but with constant intensities; (c) segmentation result of (a) with noise $\eta = 1$ and slope $h_s = 0$ added; (d) as (c) but without using scale $\sigma = 2$; (e) as (c) but $\eta = 32$ and $h_s = 32$; (f) fraction of correctly classified pixels of (a) as a function of η, for multi-scale RATS with $h_s = 0$, using 4 scales plus global threshold T_λ, compared to global thresholding according to Otsu [2]; (g) same as (f) but using just a single scale σ and T_λ; (h) same as (f) but comparing the new method to the quad-tree method [5] for different numbers of levels in the quad-tree.

The method was also tested on images of bacteria, with σ_0 ranging from 2 to 8. As can be seen in Fig. 4, when three scales are used, the method detects a faint object skipped by the quad-tree approach shown in Fig. 1. If $\sigma_0 = 2$, the method detects parts of the diffraction halos around the brighter objects.

5 Discussion

A new, multi-scale version of RATS has been developed which can adapt well to variations in both background and object intensity. A framework to select the appropriate scale has been developed. The experiments show that the method works with a modest number of scales, and with therefore a modest computational cost. A 512×512 image takes just 0.16 s if four scales are used, whereas a 2483×3508 image takes 6.03 s on a Pentium 4 at 1.9 GHz. At a single scale the timings are 0.06 s and 1.83 s, respectively. Extensions to 3-D should be straightforward, provided a 3-D equivalent of (20) is derived. The selection of the lowest scale may need more work. The experiment with synthetic images yielded the best results with $\sigma_0 = 2$, whereas the experiment using images of bacteria yielded the best results with $\sigma_0 = 4$. It might well be that the Gaussian assumptions used in (20) do not hold for small σ. Solving this problem requires analytical



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

Fig. 4. Segmentation of image of bacteria from Fig. 1(a) with $\sigma_{\max} = 16$, $\lambda = 5$ and $\eta = 2.3$: (a) $\sigma_0 = 8$; (b) $\sigma_0 = 4$; (c) $\sigma_0 = 2$; (d) contrast stretched original showing faint object and diffraction halo around brighter objects.

solutions or numerical approximations of the distribution of the denominator of (9). This will be studied in future work.

One drawback of this and many other implementations of RATS is that we need an estimate of the image noise. Ideally, we would like to be able to determine the noise from the image itself, rather than rely on external calibration data, which might be absent. If we assume that the lowest gradient pixels (or voxels) are attributable to noise only, we could estimate η by fitting the histogram of the edge filtered image at these values to the distribution in (13) in 2-D or (14) in 3-D. Work is in progress for just such an extension.

References

1. J. Kittler, J. Illingworth, and J. Föglein. Threshold selection based on a simple image statistic. *Comp. Vision Graph. Image Proc.*, 30:125–147, 1985.
2. N. Otsu. A threshold selection method from gray-level histogram. *IEEE Trans. Syst. Man Cybern.*, 9:62–66, 1979.
3. P. K. Sahoo, S. Soltani, A. K. C. Wong, and Y. C. Chen. A survey of thresholding techniques. *Comp. Vision Graph. Image Proc.*, 41:233–260, 1988.
4. Ø. D. Trier and A. K. Jain. Goal-directed evalution of binarization methods. *IEEE Trans. Image Proc.*, 17(12):1191–1201, 1995.
5. M. H. F. Wilkinson. Rapid automatic segmentation of fluorescent and phase-contrast images of bacteria. In J. Slavik, editor, *Fluorescence Microscopy and Fluorescent Probes*, pages 261–266. Plenum Press, New York, 1996.
6. M. H. F. Wilkinson. Automated and manual segmentation techniques in image analysis of microbes. In M. H. F. Wilkinson and F. Schut, editors, *Digital Image Analysis of Microbes*, pages 135–171. John Wiley and Sons, Ltd, Chichester, UK, 1998.
7. M. H. F. Wilkinson. Optimizing edge detectors for robust automatic threshold selection: coping with edge curvature and noise. *Graph. Mod. Image Proc.*, 60:385–401, 1998.
8. Y. Yang and H. Yan. An adaptive logical method for binarization of degraded document images. *Pattern Recognition*, 33:787–807, 2000.
9. I. T. Young and L. J. van Vliet. Recursive implementation of the Gaussian filter. *Signal Processing*, 44:139–151, 1995.