

A ROBUST THRESHOLDING ALGORITHM FOR HALFTONE DOTS

CHRISTINE ANTOINE¹, MIKE D. LLOYD² AND JACQUES ANTOINE³

¹previously Scientist, PAPRO, New Zealand, now with PFI, Norway

²Scientist, PAPRO, New Zealand

³Scientist, École Supérieure d'Électricité (SUPELEC), France

ABSTRACT

A robust and objective algorithm is proposed for the thresholding of halftone dot images during image processing. The algorithm is based upon detection of geometric features in the frequency histogram of grey level values. Images with a strong contrast between the halftone dots and the unprinted background area tend to have a bimodal histogram, and the threshold is defined as the position of the minimum between the two peaks in the histogram. Images with a weaker contrast between the halftone dots and the background tend to have an unimodal histogram. With unimodal histograms it is necessary to determine the point of inflexion on the histogram. The algorithm automatically selects between these two approaches. The algorithm can also detect those histograms where no threshold exists.

INTRODUCTION

The quality of halftone dots is an important determinant controlling the overall print quality in many printing applications. Quality in this context has many meanings (1,2), but important parameters include halftone dot gain, coverage, mottle and contrast with the unprinted areas. Measurement of these parameters by image analysis (3) requires the system to separate the dot areas from the unprinted areas. This step, known as thresholding, is deceptively difficult, and has a large effect on the measured dot properties. Thresholding is also used for other image analysis applications in pulp and paper research, for example determining the boundaries of solid printed

areas, measuring dirt and particles in paper (4), and determining fibre area in sheet cross-sections (5).

Halftone dot quality is typically evaluated after acquiring a magnified image of the surface of the paper. Images are typically acquired as black and white (monochrome) images, but it is also possible to acquire colour images. An 8-bit monochrome image has 256 divisions between pure black and pure white. Colour images are typically saved as three sub-images, each associated with one of the red, green, and blue colour components. Each sub-image is saved as a monochrome image, i.e, with 256 divisions between (for example) pure cyan and pure white.

These divisions are commonly called ‘grey levels’, even for the colour images. The ‘histogram’ of grey levels is a frequency distribution showing the number of pixels in the image at each grey-level. An example of a grey level histogram is shown in Figure 1. Figure 1 has two distinct peaks, and is a bimodal distribution. The peak on the left represents the grey levels of the halftone dots, while the peak on the right represents the grey levels of the unprinted background areas. The ‘threshold’ in Figure 1 can be easily defined as the grey-level where the histogram has a local minimum between the two peaks. Areas of the image with grey levels below the threshold are defined as dot areas, and areas of the image with grey-levels above the threshold level are defined as unprinted.

However, there are many histograms for which there is no clear-cut local minimum, Figure 2, due to the ink peak merging with the paper peak. These histograms are from images printed with an ink having a low contrast with the paper, for example a yellow ink, or where the percentage dot coverage was very low or very high. The absence of a local minimum means that another thresholding method is required.

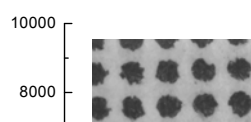


Figure 1: Bimodal histogram with corresponding halftone image (laboratory offset black print on LWC, 30% dot coverage)

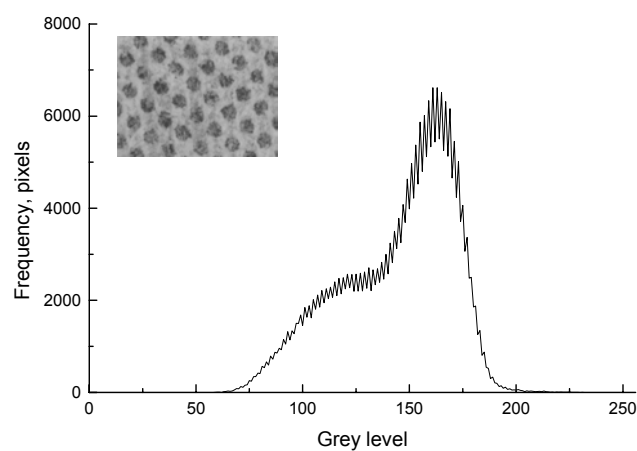


Figure 2: Unimodal histogram with corresponding halftone image (poor quality black commercial offset print on LWC paper, 20% dot coverage).

The difficulty of thresholding unimodal histograms has led to a range of thresholding algorithms being proposed for their solution, some of which have been reviewed by Barratte *et al.* (6).

Barratte *et al* themselves developed a thresholding procedure based on a series of transformations of the image. This mostly led to an image with a bimodal histogram, which was easily thresholded. Threshold values obtained using the Barratte *et al* method were found to correlate well with subjective threshold values for a wide range of images, and only failed for images with very poor contrast such as poor quality yellow prints. Such images are very hard to threshold even subjectively.

While the method of Barratte *et al* has potential, there were some problems. Unpublished investigations within PAPRO have shown that this algorithm occasionally produces a trimodal image instead of a bimodal image after the required series of erosions and dilations. Neither of the two minima in these trimodal histograms gave an acceptable thresholding on the original images. This algorithm also took several minutes to threshold each image with the PC-based Optimas image processing system that was used. This was caused by the presence of a large number of loops, which are time consuming to process when executed with a non-formatted language such as ALI used by Optimas⁶.

An alternative method is that of Otsu (7). This method divides a given image into two classes, these being the grey levels above and below a potential threshold level. Otsu suggests that the optimal threshold level is that which maximises the ‘between-class variance’ for all of the pixels in the image. The method for calculating the between-class variance is given in the Appendix. While the Otsu algorithm is fast, it still failed to give a sufficiently close match to subjectively determined threshold levels, especially for unimodal histograms.

For these reasons a new algorithm for thresholding unimodal histograms is proposed. This algorithm is a development of the principle of thresholding bimodal histograms by searching for the minimum between the two peaks. Since by definition there is no minimum on an unimodal curve, it is instead required to find the nearest thing to a minimum, namely a point of inflexion on the curve. This strategy requires a rigorous algorithm for determining the best position for the point of inflexion. This algorithm is combined with another algorithm that works better for bimodal algorithms. The advantages of this strategy is that it is possible to threshold almost all halftone dot images objectively using only the shape of the curve.

It was found that bimodal and unimodal histograms were best analysed by separate algorithms. The criteria for objectively deciding whether an algorithm is bimodal or unimodal is introduced as part of the unimodal algorithm.

Derivative algorithm for unimodal histograms

The derivative algorithm determines the position of the point of inflexion lying to the side of the main peak of the histogram, this is a method already used in other fields, e.g spectroscopy and chromatography analysis (8). The histogram actually goes through a number of steps before this. The following example is based upon the histogram shown in Figure 1.

1. The histogram is normalised (by dividing by its maximum value) and smoothed using a 4th order Butterworth filter, Figure 3. The Butterworth filter is designed to remove the high frequency noise in a signal (details of this are given in the Appendix).

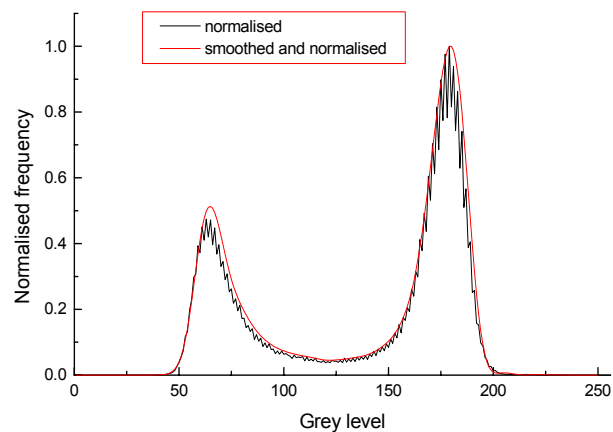


Figure 3. Example of a bimodal histogram, after normalisation and smoothing.

2. The grey levels of the beginning, end and highest point of a curve are determined. These are defined respectively as the first grey level where the normalised frequency > 0.005 (*histo_begin*), the last grey level where the normalised frequency > 0.01 (*histo_end*), and the grey level corresponding to the maximum of the curve (*imax*). The values of 0.005 and 0.01 were chosen from experience after analyzing 100 histograms.

3. The histogram is checked to see if the point of inflexion is to be found at a position to the left or to the right of the highest peak of the histogram. This is done by determining whether *imax* is closer to *histo_begin*, or to *histo_end*. In Figure 4 *imax* is closer to *histo_end*, so no further work is required. However in some histograms *imax* is closer to *histo_begin*. The rest of the algorithm is dependent on *imax* being closer to *histo_end*, so these histograms need to be ‘mirrored’. The mirrored histogram is then used as the basis of threshold detection, and the calculated threshold is subsequently adjusted so that the effect of the mirroring is removed.

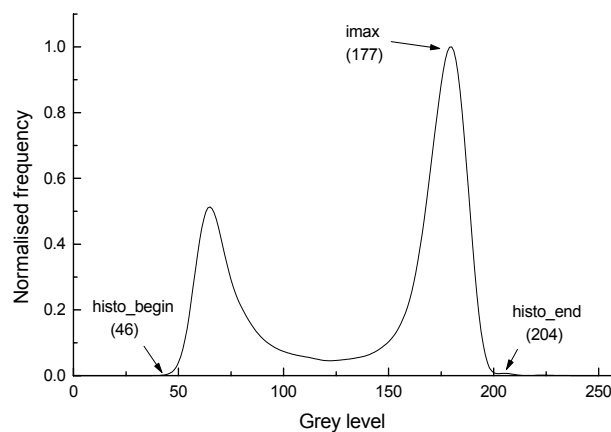


Figure 4. Example of a bimodal histogram where *imax* is closer to *histo_end*.

4. The histogram is differentiated, Figure 5.

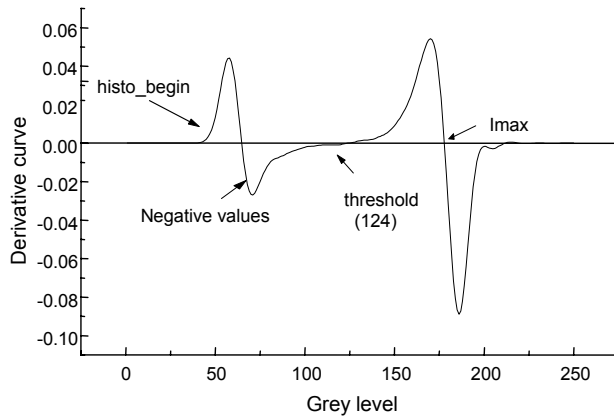


Figure 5. First derivative curve of Figure 4. The threshold is defined as the point where this curve passes through zero between *left_limit* and *right_limit*.

5. The derivative curve is checked between *histo_begin* and *imax* to see whether it goes negative at any stage. If negative values are present, the histogram is bimodal. Experience has shown that bimodal histograms are best thresholded by the bi-minimum algorithm, described in the next section. Otherwise, if the histogram is unimodal (Figure 6), the process continues.
6. The first derivative is smoothed using a running averaging formula:

$$Y(i)_{\text{filtered}} = \frac{1}{21} \sum_{i-10}^{i+10} Y(i)$$

7. Looking at the curve from *right to left*, the threshold is the X-axis value of the local minimum after the first maximum on the smoothed curve. We have defined these positions as '*left_min*' and '*right_max*', see Figure 7.

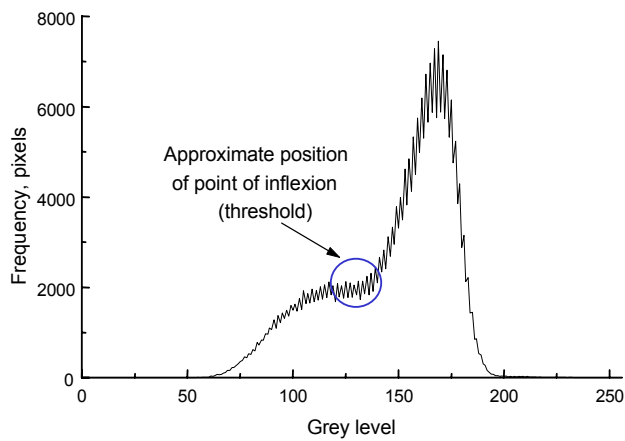


Figure 6. Example of an unimodal histogram. The point of inflexion (threshold) lies somewhere in the circled region.

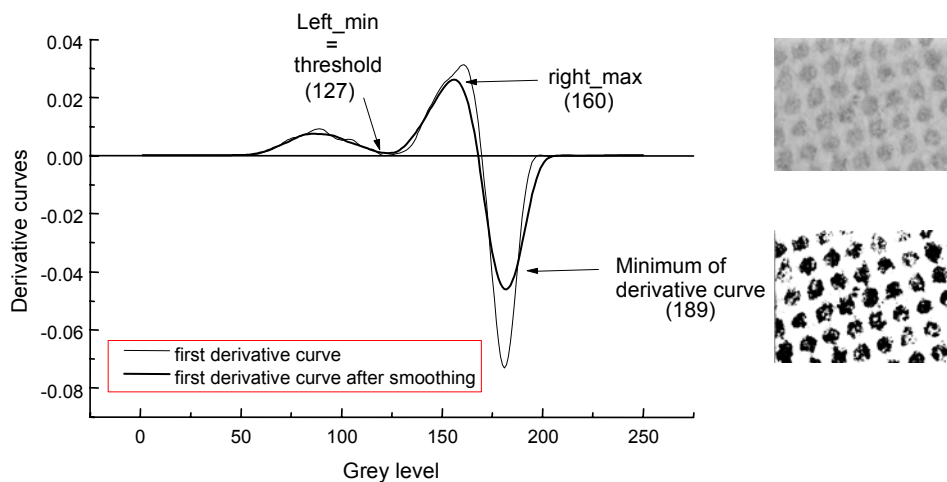


Figure 7. First derivative curve, after smoothing. This curve is then used to determine the threshold level. Corresponding original image (cyan commercial print on LWC paper, 20% dot coverage) and binary image obtained by the derivative algorithm is also shown.

Situations where using a smoothed derivative curve is inappropriate

With some unimodal histogram curves, the resultant smoothed derivative curve does not have a local minimum on the right of *histo-begin* (this is detected by the algorithm). With these

histograms it is necessary to determine the threshold by looking for left_min and right_max on the unsmoothed curve. An example of an unimodal curve where the smoothed derivative curve doesn't pass through a local minimum on the right of *histo_begin* is shown in Figure 8, and its smoothed and unsmoothed derivative curves are shown in Figure 9 below. In this case, the software can warn the user that the print and/or the image contrast is of poor quality.

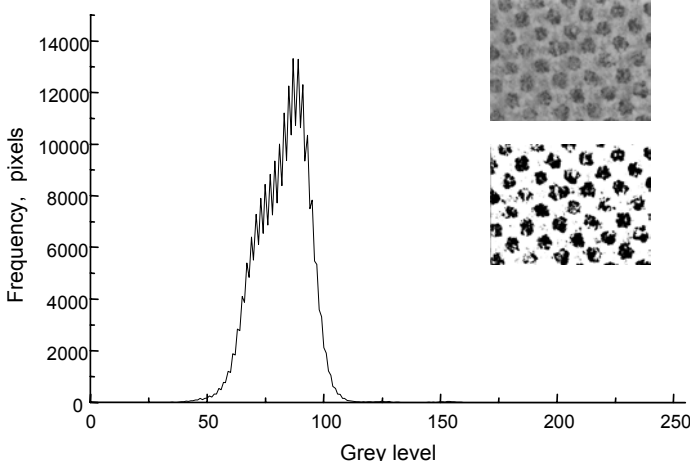


Figure 8. Another example of an unimodal histogram with corresponding original image (poor quality black commercial offset print on LWC paper, 20% dot coverage) and the binary image obtained by the derivative method without smoothing of the derivative curve.

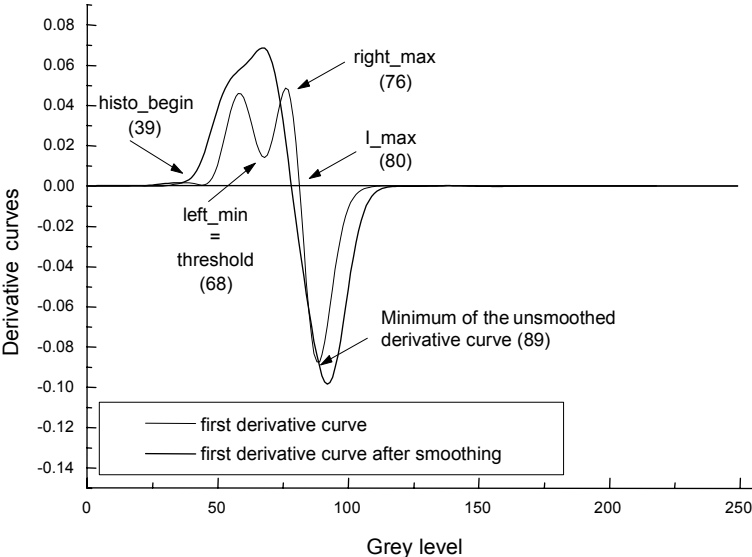


Figure 9. Derivative curves for the histogram shown in Figure 8. The smoothed derivative curve in this situation does not have a local minimum.

Histograms without a shoulder

Some histograms, for example that shown in Figure 10, do not have a shoulder, i.e. no threshold exists according to our algorithm. These algorithms can be detected as there is no local minimum *left_min* on the right of *histo_begin* on either the smoothed or unsmoothed derivative curves.

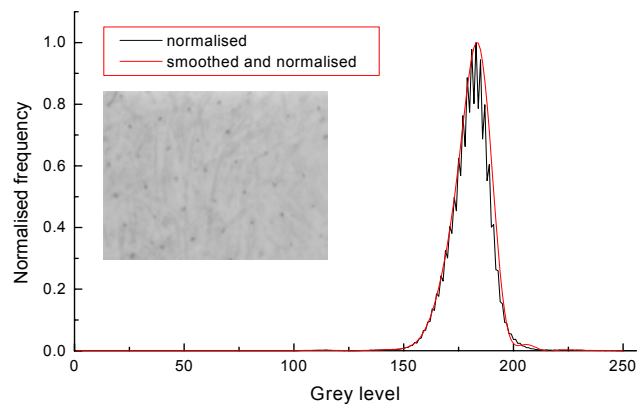


Figure 10. Example of a histogram where there is no shoulder with corresponding halftone image (cyan commercial offset print on LWC paper, 1% dot coverage).

Bi-minimum algorithm for bimodal algorithms

The bi-minimum algorithm for bimodal histograms is simple:

1. Determine the positions of the two peaks in the histogram, say *left_peak* and *right_peak* in

Figure 11 below:

- a) Use the Otsu algorithm to estimate the local minimum between these two peaks.
- b) Check this estimate of the minimum by making sure that there is no other minimum within ± 5 grey levels of the estimated local minimum.
- c) Repeat step b) starting from the estimated local minimum given by step b).

- d) Search for the maxima on either side of the final minimum determined in step c).
2. Determine the position midway between these two peaks (*'midway'*).
 3. Determine the position of the lowest point between *left_peak* and *midway*.
 4. Determine the position of the lowest point between *midway* and *right_peak*.
 5. Average the two minima determined in steps 3 and 4. This is the final threshold.

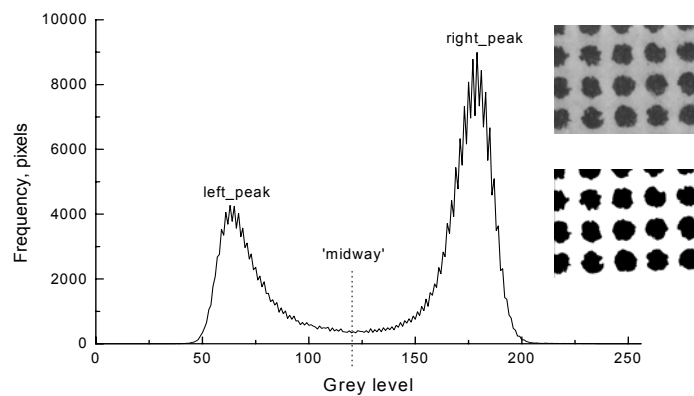


Figure 11. Definitions of *left_peak*, *midway* and *right_peak* in a bimodal histogram. Original image corresponding to this histogram (laboratory offset black print on LWC, 30% dot coverage) and the binary image obtained by the bi-minimum algorithm is also shown.

RESULTS AFTER EVALUATION OF 100 HISTOGRAMS

The sample base of 100 print images and the experimental procedures employed are described in the Experimental Section, *cf* Appendix B. All one hundred images were subjectively thresholded by three judges with printability experience. Images were thresholded by adjusting the threshold level on a screen that showed the thresholded areas overlaying the halftone dots. The judges (individually) attempted to match the thresholded areas to the halftone dot areas. The average standard deviation of the subjective threshold values (between the three judges) was 8.0.

The closeness of the match between the calculated and subjective thresholds are summarised in Table 1 below. Thresholds produced by the Otsu algorithm are compared as well. These values show a close match between the subjective and calculated thresholds. Table 1 implies that the bi-minimum threshold was, on average, the best predictor of the subjective threshold. However when evaluating these results it was checked to see whether the calculated threshold was (subjectively) ‘acceptable’, i.e. whether the thresholded areas matched the halftone dots after overlaying the dots on the computer screen. It was found that some unimodal histograms were not acceptably thresholded by either Otsu or bimodal thresholding. Therefore, when the bimodal threshold can be used, it gives a closer match to the subjective thresholding than will the derivative threshold, but it cannot be used in all cases. Moreover, the smoothing step of the derivative in the derivative thresholding was slow with our device (about 3 s) while the bi-minimum threshold is obtained instantaneously (a few ms). For this reason, it was not judged efficient to use the derivative threshold in all cases as usually a large number of images are acquired to establish paper print quality. Therefore, it was decided that the combination of the bi-minimum threshold for bimodal histograms and derivative threshold for unimodal histograms gave the best results.

Table 1. Summary of the performance of the different histograms.

Algorithm	Average (absolute) error*	Average error**
Otsu	6.3	+3.8
Bi-minimum only	4.1	+1.2
Derivative only	4.7	-2.8

Bi-minimum + derivative
(Final complete algorithm)

4.2

-0.8

Note: * 'Average (absolute) error' is defined as the average level of the absolute differences between the calculated and subjective threshold.

** 'Average error' is the average value of the figure 'calculated threshold - subjective threshold', and shows whether the calculated figure over- or under-predicts the threshold relative to the subjective threshold.

Dealing with difficult histograms

As previously discussed, some of the unimodal histograms did not have a clearly defined shoulder (point of inflexion). These histograms were from images that were also very hard or impossible to subjectively threshold. Even if a threshold could be found, the thresholded dots were very ragged and did not correspond to the dots that the operator could see on the screen before thresholding. These images were those from prints with very poor contrast (yellow prints) and/or with very low dot coverage (1 %).

Thresholding of difficult images can be improved by acquiring them as colour (RGB, or Red, Green and Blue 'channels') images rather than single channel images. Contrast is improved, and one of the three colour channels should be bimodal, or at least unimodal with a clear shoulder, allowing a threshold to be determined. The other colour channels are not thresholded. Colour image files require three times more memory however, and so will take longer to handle.

CONCLUSIONS

A robust and totally objective way to threshold halftone dot images has been developed. It is based on the detection of geometric features on the grey level histograms. Different algorithms are used to calculate the thresholds for bimodal and unimodal histograms.

ACKNOWLEDGMENTS

The authors thank Sue Williams and Maria Myohänen for their technical participation in this project as well as Stuart Corson for his useful remarks and the revision of this manuscript.

REFERENCES

1. ANTOINE, C. 1998: Print quality evaluation by image analysis. Pp. 49-55 in "Pan-Pacific and International Printing and Graphic Arts Conference".
2. BERLI, E.L. et al. 1994: The Dagbladet full-scale printing trials, Part 2: Print-quality evaluation by subjective comparison and image analysis. *Tappi Journal* 77(4): 151-159.
3. CPPA, Technical Section 1997: Basic Guidelines for Image Analysis Measurements. Useful Method PAPTAC D.30U.
4. CPPA, Technical Section 1997: Equivalent Black Area (EBA) of visible dirt in pulp, paper and paperboard by image analysis. Useful Method PAPTAC D.35.
5. DICKSON, A.R. 1999: Quantitative analysis of paper cross-sections. Pp. 735-740 in "53rd Appita Ann. Gen. Conf.". Appita, Melbourne.
6. BARRATTE, C. et al. 1993: An automatic determination of threshold for the image analysis of prints. Pp. 451-459 in "*Advances in Printing Science and Technology: Proceedings of the 23rd Research Conference of IARIGAI, Paris, September 1995*". John Wiley and Sons, London.
7. OTSU, N. 1979: A thresholding selection method for grey-level histograms. *IEEE Trans Systems, Man and Cybernetics* SMC-9(1): 62-66.
8. DYSON, N. 1990: Chromatographic integration methods. Series Editor, Roger M. Smith, University of Technology, Loughborough, UK.

APPENDIX - A

Otsu thresholding algorithm

The Otsu algorithm (6) determines the threshold by determining the grey level ' k ' that maximises the 'between-class variance' $\sigma_B^2(k)$ of the grey level histogram, where:

$$\sigma_B^2(k) = \frac{[\mu_T \omega(k) - \mu(k)]^2}{\omega(k)[1 - \omega(k)]} = \text{between class-variance of histogram}$$

[2]

where:

$$\mu_T = \sum_{i=1}^{256} i p(i) = \text{mean value of histogram} \quad [3]$$

$$\omega(k) = \sum_{i=1}^k p(i) = \text{zeroth-order cumulative moment of the histogram} \quad [4]$$

$$\mu(k) = \sum_{i=1}^k i p(i) = \text{first-order cumulative moment of the histogram} \quad [5]$$

$$p(i) = n(i)/N = \text{normalised (probability) density function of histogram} \quad [6]$$

$n(i)$ = number of pixels at grey level i in original histogram.

k = potential threshold (grey level)

N = total number of pixels in histogram.

'class' is defined as the groupings of grey levels either above or below a given k .

Butterworth filter function

Unimodal histograms were smoothed using a 4th order lowpass Butterworth filter with a cut-off frequency of 1/8. The filter vectors A and B were designed using the ‘Butter’ function in Matlab mathematical software as [1, -2.98, 3.42, -1.79, 0.356], and [9.33*10⁻⁴, 3.73*10⁻³, 5.60*10⁻³, 3.73*10⁻³, 9.33*10⁻⁴] respectively.

The filtered (after normalisation) histograms were calculated using the three part algorithm:

$$1) \ t(i) = 0 \quad (i = 0 \text{ to } 256) \quad [7]$$

$$2) \ t(i) = \sum_{j=1}^5 B(j)h(i-j+1) - \sum_{k=2}^5 A(k)t(i-k+1) \quad (i = 5 \text{ to } 256) \quad [8]$$

$$3) \ f(i) = t(i + 6) \quad (i = 7 \text{ to } 256) \quad [9]$$

where $h(i) = n(i)/\max[n(i)]$ = normalised value of histogram at grey level i [10]

$t(i)$ = temporary holding variable.

$f(i)$ = value of normalised histogram at grey level i after smoothing.

APPENDIX – B

EXPERIMENTAL

The PAPRO image analysis system and calibration routine used for this experiment has been described earlier (1). Images are captured through a lens and a CCD video camera. Illumination of the samples is provided by a ring light. Images are analysed with a frame grabber, computer and image analysis software. The acquired image areas are 1.8 by 2.4 mm in size with a resolution of 266 pixels/mm. Thus, about 10 to 20 halftone dots are present on each image. The actual number of dots depends on the screen ruling. The number of pixels used to represent each dot depends on the screen ruling and the dot coverage. A JVC, colour, CCD, video camera provides high horizontal resolution of 750 lines and an effective number of 440 000 pixels. The computer is a standard PC clone, 200 MHz Pentium pro-processor with 64 MB of memory. Windows 95 is the operating system. Optimas 6.0 is the image analysis software. Matlab version 5.3 (*aka* Release 11) was also used during the development of our algorithm.

The thresholding routines, that are the subject of this paper, were developed and confirmed by comparing the calculated threshold levels to subjectively-obtained threshold levels for a diverse series of one hundred halftone dot images. These images were chosen to represent a wide range of halftone dot images in terms of printing colour, contrast between basepaper and printing colour and the dot to ensure the universal applicability of the developed algorithm.

The images were printed on newsprint, LWC and bleached and unbleached kraft liners from different origins. They were printed on an IGT F1 laboratory press, by the flexographic process for liners (1), or by a waterless offset process developed at PAPRO. LWC samples from a commercial heatset, offset printing run were also included in the sample set. Screen ruling were 70 lpi and 100 lpi for conventional screening. Stochastic screenings were also studied. Dot coverage ranged from 1% to 80%. Black, cyan, yellow and magenta ink were used for the different laboratory and commercial printing processes.

(1) Chalmers, I. R.: Flexographic printability of packaging-grade papers. *Appita*. 51 (3), 193, May, 1998.