EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling (2004)
John F. Hughes and Joaquim A. Jorge (Editors)

Adaptation of Brensen's Thresholding Algorithm for Sketched Line Drawings

A. Bartolo[1], K. Camilleri[1], J.Borg[2] & P. Farrugia[2]

University of Malta, Malta
[1] Department of Microelectronics
[2] Department of Manufacturing Engineering

Abstract

Image binarisation is one of the first image processing techniques within a sketched line drawing interpretation system. In order to achieve an automated system, it is necessary to have a single algorithm which may be used for all image types. This is difficult to obtain if the parameters required by an algorithm are set manually, however, the adaptive evaluation of parameters form image properties is a step towards this goal. This paper discusses the methods by which the parameters required for Brensen's Algorithm may be evaluated adaptively form the image properties.

Categories and Subject Descriptors (according to ACM CCS): I.4.3 [Image Processing and Computer Vision]: Pixel Classification

1. Introduction

In the design process, designers prefer the use of the paper-based sketch at the initial design stage, as this provides a higher degree of flexibility due to its portability. In order to integrate the paper-based sketch with existing Computer Aided Design (CAD) tools, these sketches must be converted into a digital format. This may be achieved by means of flatbed scanners or digital cameras as shown in Figure 14. However, the resulting digital representation is a low-level image and cannot be used directly by the CAD tools. This is especially so when sketches are digitised by means of camera-phones which introduce variant light intensities to the digital image. Thus, an interface between the paper-based sketch and these software tools is required in order to extract the relevant information from the digitized image and represent the sketched drawing in a format that may be interpreted by the CAD tools.

The required interface may be obtained through a line drawing interpretation system [Seg03] that identifies the sketched drawing from its background and represents the object in terms of vectors, which may be correctly interpreted by the CAD tools. The accuracy of the interpretation of the sketched line drawing is limited by the accuracy of binarisation techniques, which identify the image foreground from its background. It is therefore necessary to obtain a reliable and robust binarisation method that gives the possibility of a correct extraction of the sketched line drawing from its background.

This section introduced the sketched line drawing interpretation system and the need for a suitable binarisation technique. This paper is based on the study of Brensen's binarisation algorithm and the adaptations required to overcome its limitations. This section is followed by a review of different binarisation algorithms and their relevance to sketched line drawings, including the advantages of Brensen's algorithm over these algorithms. Section 3 further explains Brensen's algorithm, leading to the modifications and adaptive parameter evaluation, which enhance the performance of the algorithm in Section 4. Section 5 introduces a method by which shadow boundaries may be removed from binarised images when, due to the nature of the shadow boundary, the original algorithm fails. A discussion of the results obtained is presented in Section 6, followed by Section 7 which presents the conclusions drawn from this work.

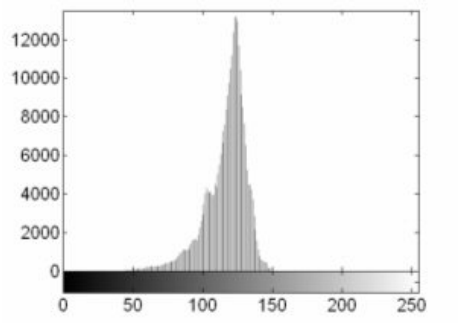

Figure 1: *Histogram of grey level distributions for the image shown in Figure 14a.*

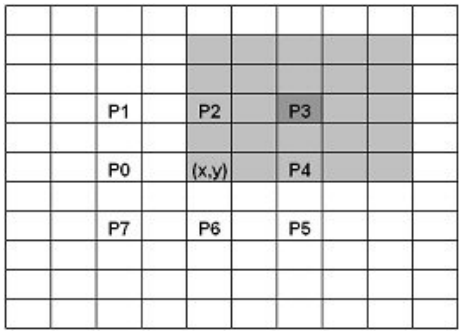

Figure 2: $(2SW+1)\times(2SW+1)$*-size neighbourhood centered at point P3 in the neighbourhood of the pixel in position* (x,y).

2. Background Work

The process of binarisation is used to distinguish between foreground and background pixels by assigning a class label to each pixel within the image. Thus, a grey level image will be expressed in terms of two grey levels. Binarisation techniques may be divided into a number of categories, and the selection of a suitable technique should reflect the properties of the image being binarised, since techniques that are suitable for one class of images are not necessarily suitable for other image types. It is therefore necessary to examine the properties of the sketched line drawing in order to identify the binarisation techniques that are most suitable for this application.

2.1. Properties of sketched line drawings

Sketched line drawings are carried out by designers during the process by which the object geometry is being defined. Thus freehand sketches are often made up from a number of strokes of different lengths, thickness and intensity [AP00]. As illustrated in Figure 14, the sketched drawing is likely to occupy a large area of the paper, however since sketches are drawn by using long thin line strokes, the number of foreground pixels is much smaller than the background pixels. Furthermore, the image background is likely to contain a higher variance due to inhomogeneous paper and different light intensities. Thus a histogram of the grey-level distributions will show a large peak corresponding to background grey-levels and a smaller peak corresponding to the foreground grey-levels as shown in Figure 1. This makes it difficult to establish a single threshold which will correctly identify the pixel class. For these reasons, local binarisation techniques are more adapted for sketched line drawings since a threshold is evaluated for each pixel within the image according to the properties of the neighbouring pixels.

2.2. Review of some local binarisation techniques

This section describes four binarisation algorithms namely those proposed by Kamel and Zhao, Eikvil, Niblack and Palumbo and Guliano. These techniques have been cited in literature as being suitable for line drawings and text images, which exhibit the same characteristics as sketched line drawings.

Kamel and Zhao's [KZ93] logical adaptive technique compares the grey level of the pixel in consideration with eight local averages in the $(2SW+1)\times(2SW+1)$ size neighbourhoods centred at the four points $P_i, P_i^{'}, P_{i+1}, P_{i+1}^{'}$ as shown in Figure 2, where $P_i^{'} = P_{(i+4) \bmod 8}$ and SW represents the stroke width of the line drawing. A comparison operator is derived from these averages and is used to determine the class of the pixel in consideration.

The algorithm requires two user defined parameters, namely the stroke width SW and an initial threshold T which is used to evaluate the required comparison operator. Yang and Yan [YY00] propose methods by which these parameters may be calculated adaptively. However, the adaptive evaluation of the parameter Trequires another parameter α. Yang and Yan specify a range of values of α for which suitable values of T may be obtained. However, this range was specified for text images having only variable illumination in the background. A more suitable range of values for α has been identified [Bar04] such that the value of T may also reflect inhomogeneous backgrounds.

Thus, the performance of this algorithm depends on the parameter α but the algorithm is less sensitive to this parameter which may be set more easily than the parameter T.

Eikvil's method [TJ98] makes use of two concentric windows L and S and Otsu's thresholding technique [GW02] to determine the class of the pixels in the smaller window S. The pixels within the larger window L are temporarily classified into two classes by Otsu's threshold. The mean of these two clusters is evaluated and their difference is compared to

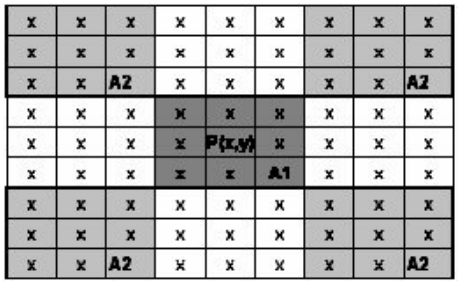

Figure 3: *The five* 3×3 *local pixels within the* 9×9 *used to evaluate Palumbo and Guliano's threshold.*

a parameter k which will determine if there is sufficient contrast between the two clusters. This will determine the effectiveness of Otsu's threshold. Thus, if the difference between the two means is larger than k, then the pixels within the smaller window are thresholded according to Otsu's threshold. Otherwise, the pixels are assigned to the class whose label is closest to the mean grey level within the smaller window S.

Hence, this algorithm is dependent on three parameters namely, L, S and k whose values must be specified manually. The algorithm shows least sensitivity to the smaller window size S, which may be fixed at 3×3 since this is the smallest window size possible. However, suitable values of parameters L and k must be determined for each image. Since the pixels within the larger window L are used to evaluate Otsu's threshold, the size of this window cannot be determined in the same manner as that of Kamel and Zhao's algorithm described previously. Furthermore, this parameter cannot be fixed to a particular value since it must reflect the local characteristics of the image and is therefore dependent on the image properties. The algorithm is most sensitive to the parameter k which will determine whether the pixels in the smaller window S are classified according to Otsu's threshold or if they are assigned to a single class label. Thus k strongly affects the performance of the algorithm.

Niblack's algorithm [BM04] evaluates a threshold for each pixel within the image according to the mean and standard deviation of the pixels in a predetermined window W centred at the pixel in consideration. The pixel class is determined as follows :

$$w(x,y) \rightarrow \begin{cases} w_f, \ p(x,y) < \ T(x,y) \\ w_b, \ p(x,y) \geq \ T(x,y) \end{cases} \tag{1}$$

where, $\mu(x,y)$ is the mean grey level of the pixels within the window, $s(x,y)$ is the standard deviation of these pixels and k is a user defined parameter which adjusts the amount of print object boundary that is taken as part of the foreground. The performance of the algorithm is therefore dependent on the window size and the parameter k. Analysis of the image properties shows that the value of k should be small or negative [Bar04]. However, optimal values for each image must be established since k has a direct effect on the evaluated threshold. Palumbo and Guliano's [YY00] differs form the previous algorithms since it uses a fixed 9×9 window to evaluate the class of each pixel within the image. The pixel value is determined according to the five 3×3 local pixels within the 9×9 window as shown in Figure 3. An initial threshold is used to determine the pixels which definitely belong to the background whilst the remaining pixels are classified using a more complex label assignment rule. A second threshold T_2 is defined and used to threshold the regions labelled A_2 in Figure 3. The average μ_2^b of the pixels within these regions which have a grey level greater than T_2 is calculated. If the average grey-level of pixel in A_1 is represented by μ_1, pixel label assignment is determined according to:

$$w(x,y) \rightarrow \begin{cases} w_f, \ T_4\mu_1 < T_3\mu_2^b + T_5 \\ w_b, \ otherwise \end{cases} \tag{2}$$

Thus, although the algorithm makes use of a fixed window size, it requires five separate parameters of which, only $T1$ and $T2$ have any particular physical meaning. $T1$ represents an initial global threshold of the image whilst $T2$ represents a local threshold value. Whilst the values of the remaining parameters do not have a direct physical meaning, their value is of direct significance in the thresholding process since they are used to determine the class of the image. This makes the algorithm highly dependent on the parameters T_3, T_4 and T_5.

Thus, although different methods of image thresholding exist which may be applied to sketched-line drawings, these methods all require some user defined parameters. Thus, the algorithms studied are unsuitable for application to an automated sketched-line drawing interpretation system. Kamel and Zhao's technique shows least sensitivity to the required parameters, however the requirement for the single parameter α hinders the automation process. Brensen's algorithm is similar to Eikvil's algorithm but uses a single window whose size is easier to evaluate than that required by Eikvil. It also makes use of a parameter k to determine the contrast within this window. Unlike the parameters used by Niblack and Palumbo and Guliano, the behaviour of this parameter may be directly observed from the images in consideration. This makes it possible to observe the nature of the parameter and hence determine its value.

3. Review of Brensen's Algorithm

Let I represent an $R \times C$ image with, $I[x_c, y_c] \in 0, 255, I \leq x_c \leq R, 1 \leq y_c \leq C$ where 0 denotes a black pixel, considered as foreground and 255 denotes a white pixel, considered as background. A threshold T is calculated for each pixel with coordinates (x_c, y_c) from the properties of an $r \times r$ window W where $x_c - \frac{r}{2} \leq x \leq x_c + \frac{r}{2}$ and $y_c - \frac{r}{2} \leq y \leq x_c + \frac{r}{2}$. A label $\omega(x_c, y_c)$ is assigned to each pixel according to the threshold $T(x_c, y_c)$. A pixel identified as a background pixel

is assigned a label ω_b whilst a foreground pixel is assigned a label ω_f . The algorithm proposed by Brensen [TJ98] defines the threshold $T(x_c,yc)$ as the mean of the maximum and minimum grey levels within the window. If the window W is centred at pixel (x_c,y_c), the threshold for $I(x_c,y_c)$ is defined as:

$$T(x_c,y_c) = \frac{1}{2}(Z_{\max} + Z_{\min}) \tag{3}$$

However, this threshold gives satisfactory results only when the contrast $C(x_c,y_c)$, defined as:

$$C(x_c,y_c) = Z_{\max} - Z_{\min} \tag{4}$$

is sufficiently large. Brensen argues that when this contrast is less than a specified value k, all the pixels within the window $W(x,y)$ may be set to background or foreground according to the class that most suitably describes the window. This may be obtained by comparing the mean grey level within the window with the threshold $T(x_c,y_c)$ as shown in Equation (4).

Thus if $C(x_c,y_c) \geq k$,

$$w(x_c,y_c) \rightarrow \begin{cases} w_f,\ I(x_c,y_c) > T(x_c,y_c) \\ w_b,\ I(x_c,y_c) \leq T(x_c,y_c) \end{cases} \tag{5}$$

whereas if $C(x_c,y_c) < k$

$$w(x_c,y_c) \rightarrow \begin{cases} w_f,\ \mu(x,y) > T(x_c,y_c) \\ w_b,\ \mu(x,y) \leq T(x_c,y_c) \end{cases} \tag{6}$$

Thus, the performance of this algorithm is strongly dependent on the value of k. This parameter must be large enough such that $C(x_c,y_c) < k$ is satisfied for all the windows having only background pixels. In this way, individual pixel labelling as given by Equation (5) is applied only to windows containing a mixture of foreground and background pixels.

Brensen's algorithm does not evaluate a value for k and r, hence these must be determined by trial and error for each image in order to achieve suitable results. This is required since the different image properties lead to different optimal values for k. Although a range of k values may be found suitable for a particular image, this range depends on the quality of the background and a common range of suitable values cannot be obtained as illustrated in Table 1.

Image	k values	r
Figure 14a	$30 \leq k \leq 45$	8
Figure 14b	$30 \leq k \leq 45$	8
Figure 14c	$15 \leq k \leq 25$	4
Figure 14d	$35 \leq k \leq 58$	16
Figure 14e	$20 \leq k \leq 40$	4

Furthermore, the parameter k is also dependent on the window size r since this determines the sample of pixels from which the maximum and minimum grey levels are obtained. It is therefore necessary to evaluate adaptively the parameters k and r.

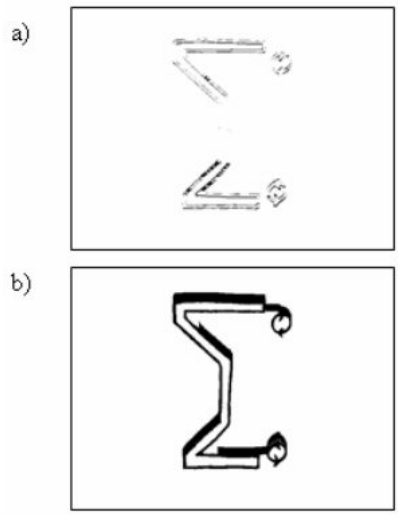

Figure 4: *Illustrates the effect of the window size r on the classification of pixels. a) $r = 4$ shows misclassifications of the foreground pixels b) $r = 30$ shows misclassifications of some of the background pixels. In both cases, $k = 40$.*

4. Modifications and Parameter Evaluation

In order to optimize the algorithm to suit the sketched line drawing application, analysis of the properties of sample drawings is necessary. Since sketched line drawings consist of long thin line structures, it is highly unlikely for a selected square window to contain only foreground, or line drawing, pixels. If large dark regions are present in the sketched drawing, these are often due to shadows caused by variable illumination when an image is captured. These large dark areas are also considered as background regions.

Thus, small contrast values indicate that the pixels within the selected window are all background pixels. Hence for this application, the classification of pixels represented by Equation (6) may be reformulated as:

$$w(i,j) \rightarrow w_b,\ C(x,y) < k \tag{7}$$

4.1. Evaluation of the window size r

The accuracy with which the algorithm classifies pixels depends on the size of the windows being used. If windows are too small, then the assumption that windows do not consist of foreground pixels only does not remain valid. Thus, Equation (7) would misclassify foreground pixels as background as shown in Figure 4(a). On the other hand, selecting very large window sizes implies that the local character-

istics of the neighbourhood of the pixel are lost. Thus, the window may capture grey level contrasts that are not in the neighbourhood of the pixel in consideration. This affects the values of $T(x_c, y_c)$ and $C(x_c, y_c)$, which may result in misclassification of foreground and background pixels. Figure 4(b) shows how large window sizes may cause misclassification of the background pixels which are close to the image foreground.

The size parameter r of the window **W** should be determined from the stroke-width of the image. This is a measure of the width of the sketched strokes which form the image and will therefore indicate the size of the window required such that it is unlikely for a window to contain foreground pixels only. The selection of higher window sizes is undesirable since this reduces the local characteristics of the windowed region.

Evaluation of the stroke width may be carried out by using the method proposed by Yang and Yan [YY00] in the evaluation of Kamel and Zhao's algorithm. This algorithm makes use of run-length histograms on partial areas in the image. The image is first divided into $N \times N$ regions where $N = 4,5,6,7,8$ in attempt to find some local area with a quasi-bimodal local histogram or higher local contrast. The regions are selected in the two diagonal directions if N is even and in the two diagonal, horizontal and vertical directions if N is odd, as shown in Figure 5.

The run-length information extracted from these regions is then used to evaluate the average stroke width of the image. The run-lengths, expressed as $R(i), i \in J, J = 1,2,...,L$, where L is the longest run measured and $R(i)$ the frequency of the run-length i, are calculated in the horizontal and vertical directions. However, the directions over which the run length is calculated may be increased to four directions which will then include the vertical, horizontal and diagonal runs across the selected regions.

The stroke width of the given image is then defined as the run-length having the highest frequency excluding the unit-run and may be expressed as:

$$SW = i,\ arg\{max_{i \in J}\ \{R(i),\ i \neq 1\} \qquad (8)$$

This will reflect the average width of strokes in a document image. If an image contains some complex background patterns, then it is possible that the highest peak will be formed by these factors rather than the line drawing itself. This gives rise to the need of using only the selected regions as well as black runs only to determine the stroke width.

However, stroke-width analysis is being carried out on grey-scale images which have variable illumination and are generally of poor quality. Thus, as shown in Figure 1, the number of pixels having a grey-level of 0, which corresponds to black pixels, is very small. Thus, in order to apply Yang and Yan's stroke-width evaluation method to these images, it is necessary to perform a preliminary thresholding on the image before the stroke width may be calculated

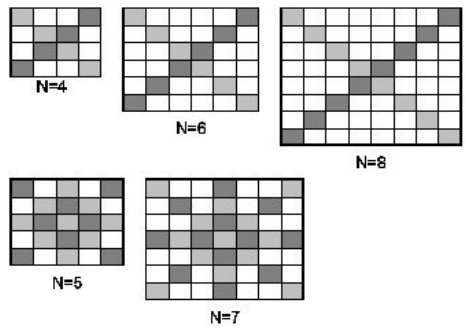

Figure 5: *The selected regions on which local region histogram analysis will be performed in attempt to find some region with quasi-bimodal local histogram or higher contrast.*

This was achieved by applying a global thresholding technique to selected regions within the image. In order to achieve suitable black runs, the region with higher contrast from each of the regions analysed for each value of N was selected. Thus, five local regions, one for each N value was used to evaluate the black runs. This results in five possible stroke widths whose average may be determined.

Since run lengths are calculated in the horizontal and vertical directions, the stroke width will reflect the length and width of the strokes in the region in consideration. For example, run lengths obtained from a region having a horizontal line would reflect the length of the line in the horizontal run-length and its width in the vertical runlength. Thus, the average stroke width would reflect a measure of the length and width of a line in the selected regions. This ensures that the window size selected is large enough to contain a suitable representation of the line drawing without loosing the local characteristics of the window.

Figure 6 shows the binarised result of the image shown in Figure 14(a) for which the stroke width has been determined adaptively. The stroke width calculated for this image is 8. This is large enough to allow proper classification of foreground pixels without causing misclassifications of background pixels in the vicinity of the foreground regions. The stroke widths obtained for the images shown in Figure 14 are listed in Table 1. These show that the highest stroke widths are obtained for images having dark shadows such as Figure 14(c) or textured backgrounds as shown in Figure 14(d).

4.2. Evaluation of the constant k

The performance of the algorithm is also sensitive to the parameter k. This determines whether the contrast $C(x_c, y_c)$

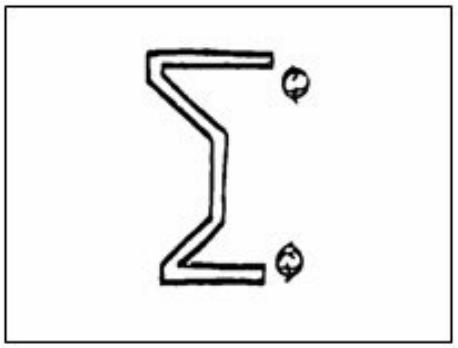

Figure 6: *Binary representation of image shown in Figure 14(a) after the window size has been determined adaptively using Yang and Yan's method. In this image, $r = 8$ and $k = 40$. The value of k has been determined by trial and error until a suitable value was obtained.*

within the window is large enough to apply the threshold $T(x_c, y_c)$ to the pixel in consideration. Since the threshold $T(x_c, y_c)$ gives accurate results for regions with high contrast, it is necessary to select a value of k such that most of the background pixels may be classified according to Equation (6). Selecting small values for k implies that $T(x_c, y_c)$ is applied to some background regions, thus, pixels with grey levels larger than the threshold are wrongly classified as foreground as shown in Figure 7(a). On the other hand, large values of k require that the contrasts within a window be large for pixels to be classified according to the value of $T(x_c, y_c)$. This results in misclassifications of the foreground (line) pixels as shown in Figure 7(b).

The proper selection of the k value depends on two image properties, namely the type of image background and the quality of the image foreground. If an image background contains some patterns or textures, such as the grid illustrated in Figure 14(a) and the textured paper in Figure 14(d), the value of the parameter k must be larger than that chosen for images having a homogenous background. This is necessary such that the value of k is sufficiently large to capture the contrast between the background pixels.

The quality of the image foreground constrains the largest value of k possible such that the contrast would not be large enough to classify windows having a mixture pixel classes into an all background window. Images in which the difference between the grey levels of the foreground and background is low, require smaller values of k than if the difference were larger. This is required so that the threshold $T(x_c, y_c)$ is applied to all windows having mixtures of foreground and background, thus in order to establish a value for k, it is necessary to examine the variation of the contrast within suitably sized windows across the image. A measure of the success with which the inequality is satisfied is of particular interest since this identifies the number of times an

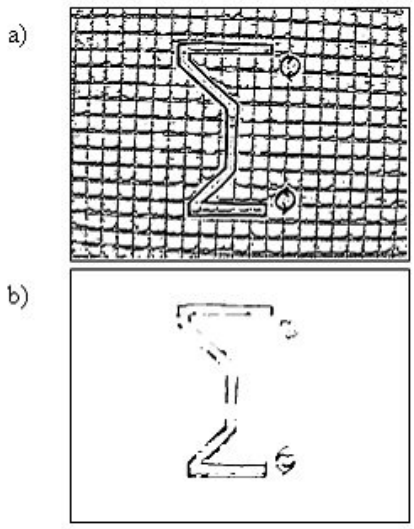

Figure 7: *Illustrates the effect of parameter k on the classification of pixels. a) $k = 10$ shows misclassifications of the background b) $k = 50$ shows misclassifications of the foreground. In both cases, $r = 8$.*

entire window is labelled as background. Figure 8 shows the plots of the number of success with different k values for the images shown in Figure 14(a) and 1(c). These curves illustrate how the number of successes is initially small since for small values of k, the inequality is satisfied only for pixels whose grey-level intensities are similar. This would therefore capture only a small number of background regions due to the variations within the background.

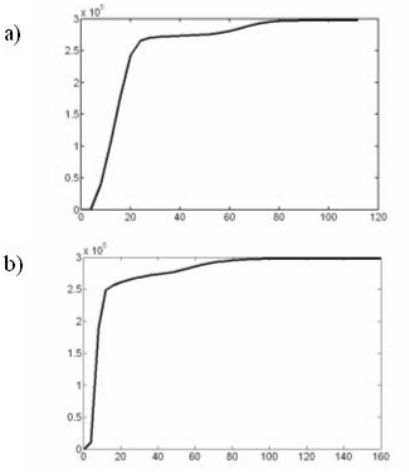

Figure 8: *Graph illustrating the number of times the inequality $C(x,y) < k$ is satisfied for a range of k values for the images shown in Figure 14(a) and (c).*

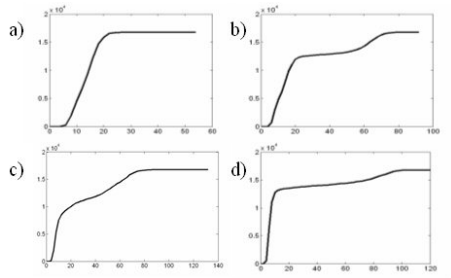

Figure 9: *Plots of success-rate against k values for (a) top left region (b) central region of the grid image shown in Figure 14(a). (c) top left region (b) central region of the dark-shade image shown in Figure 14(c).*

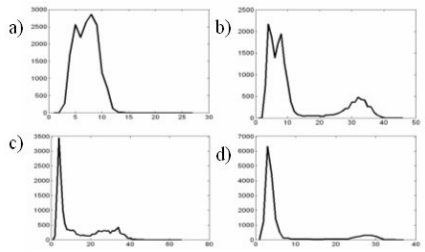

Figure 10: *Plots of the gradient of the curves shown in Figure 9.*

As k increases, the number of successes increases until the value of k is large enough to capture most of the variances within the background. When this value is reached, the number of successes will stabilize and the rate of increase in successes with respect to k will diminish, resulting in a flattening of the characteristic. However, further increase in k will show an increase in the success rate since at some value of k it would be sufficiently large to capture the contrast present between the foreground and background pixels as well.

4.3. Local evaluation of the parameter k

Images digitised by means of digital cameras can be subject to variant light intensities, therefore the contrast levels across the image will fluctuate accordingly. Thus, although the value of k has been calculated adaptively, its value will not necessarily ensure good classification of pixels, especially when foreground pixels are located in dark shadow regions. It is therefore necessary to obtain values of k which reflect the different light intensities present within an image. This may be achieved by dividing the image into smaller regions and calculating a different value of k for each of the regions. In this analysis, the image has been divided into sixteen equal sized regions. Since the value of k is determined from the contrast within a window $\mathbf{W}$, smaller regions would not provide sufficient data points for a robust estimation of k.

Figure 9 illustrates the success-rate plots obtained from the top left hand region and central region of the images shown in Figure 14(a) and 14(c). These illustrate the difference in the success-rate for different regions within the image. The value of k for which the gradient is at a minimum, may be located by evaluating the gradient of the success-rate curve at each point as shown in Figure 10.

In the general case, the gradient of regions having only background pixels will show a single maximum point followed by small gradient values since the rate of increase in the successes will stabilise for small values of k due to the small contrast present in the region. This distinguishes background regions from regions having a mixture of pixel class as the gradient of the success-rates for such images will show a large peak, followed by a minimum point and a second maximum. The first maximum point corresponds to the initial increase in the success-rate until the k value is large enough to capture the contrast between the background pixels. The second maximum point results from the fact that the value of k is large enough to capture the differences between the foreground and background pixels. The minimum point located between these two maxima indicates a value of k such that k is larger than the contrast between the background pixels but smaller than the difference between the background and foreground grey levels.

Therefore, a suitable value of k may be obtained by considering the minimum point occurring after the first maximum point. In regions having only background pixels, this would correspond to the value of k for which all pixels may be classified as background by Equation (7). In regions having a mixture of foreground and background, this implies that Equation (5) is applied only to the windows centered on foreground pixels whilst classifying all other pixels as background according to Equation (7).

5. Shadow Boundary Removal

The main limitation of this algorithm is its treatment of shadow boundaries. Since pixels lying on a shadow boundary separate regions of dark grey levels form others with light grey levels, windows lying on the shadow boundary will have sufficient contrast such that the value of $C(x_c, y_c)$ will be greater than the chosen value of k. This implies that these pixels will be interpreted as being part of the image foreground.

However, the properties of the shadow boundary pixels may be used to differentiate shadow boundary pixels from line-drawing pixels. Analysis of line drawings shows that the pixels located across the line drawing will have similar grey-level intensity. Thus, shadow boundary pixels, which separate two regions of relatively high contrast, located perpen-

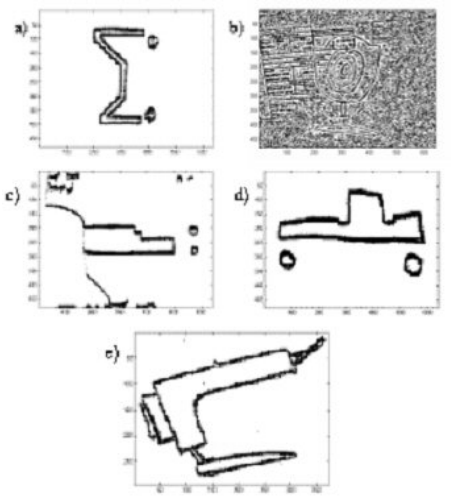

Figure 11: *Illustrates the results obtained after the images in Figure 14 are binarised using the adaptive evaluation of r and k. A single value of k is applied to the entire image.*

dicularly across the shadow boundary may be distinguished from line drawing pixels as shown in Figure 15.

In order to achieve this distinction, an approximate orientation of the line drawing is obtained by using Sobel edge detection [PM99]. A sample of pixels from the original image located on either side of pixels identified as foreground are taken into consideration and their mean values μ_1 and μ_2 are compared to the mean μ_3 of a 3×3 window centered on the pixel in consideration. If the difference between μ_1 and μ_2 is greater than any one of the differences between μ_3 and μ_1 or μ_3 and μ_2, then the pixel may be classified as a shadow boundary pixel since it is located within an area of high background contrast. In order to achieve suitable results, the pixel sample is taken at a distance of $\frac{SW}{2}$ as this ensures that the sample values do not overlap part of the line drawing but are outside the window range.

6. Results and discussion

The results obtained must be analysed with respect to two features, namely the validity of the window size and the validity of the parameter k which have been evaluated adaptively. The window sizes obtained for the images of Figure 14 shows how these window sizes may maintain the local characteristics of the image whilst being large enough to ensure that a window will never contain only foreground pixels. Larger window sizes are obtained for images having shadows or textured background. This occurs because the preliminary threshold used to identify the black runs within the selected regions, may causes some misclassifications when identifying the foreground pixels. This creates run-lengths which are larger than the actual line widths. However, the evaluation of the parameter k would require larger windows for these image types in order to identify the contrast existing within the background regions.

Figure 11 shows the initial results achieved in this analysis. These illustrate the binary representation of the images when both values of r and k have been evaluated adaptively. However, a global k rather than a number of local k values have been used. Comparison of the value of k obtained adaptively with the values determined by trial and error for each of the images shown in in Figure 14, shows how the adaptive evaluation of the parameter k identifies the best k values for the image under test. This comparison is shown in Table 2. This shows how the evaluated values of the parameter k correspond to the range of values of k which yield suitable results. This is true for all images except for that of Figure 14(c) whose result is shown in Figure 11(c). This shows the need for a regional based evaluation of the parameter k as has been defined in Section 4.3. This is further emphasised in the result obtained for Figure 11(b) in which the background pixels are misclassified as foreground due to the patterns present in the background as well as the variant light intensities. This image has been drawn with H-type pencils and hence the grey level intensity of the foreground is at times equal to that of the background. Hence the selected k value is more likely to identify the contrasts within the background rather than those present between the foreground and background.

Image	estimated k values	evaluated k values
Figure 14a	$30 \leq k \leq 45$	36
Figure 14b	$30 \leq k \leq 45$	28
Figure 14c	$15 \leq k \leq 25$	44
Figure 14d	$35 \leq k \leq 58$	56
Figure 14e	$20 \leq k \leq 40$	36

Table 2: *The estimated range of values of k and the evaluated value of k for the images shown in Figure 14.*

Figure 12 illustrates the results obtained after evaluating the value of k for local regions within the image. These results show that the local evaluation of the parameter k allows for a greater accuracy in the classification of the pixel class. This may be observed from Figure 12(c) where the image foreground located in the shadow region has been classified correctly in contrast to the results shown in Figure 11(c). Furthermore, Figure 12(b) shows how the local k values allow for the classification of the foreground form its background.

As explained in Section 5, the algorithm fails to classify shadow boundaries as background when these are present in the image. These may be removed by applying the technique proposed in Section 5. The results of the shadow removal

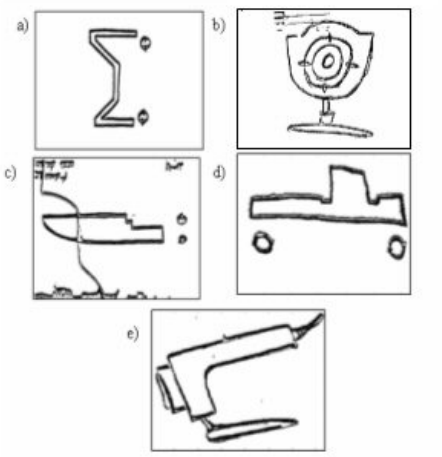

Figure 12: *Illustrates the results obtained after the images in Figure 14 are binarised using the adaptive evaluation of r and k. The parameter k was evaluated for local regions within the image.*

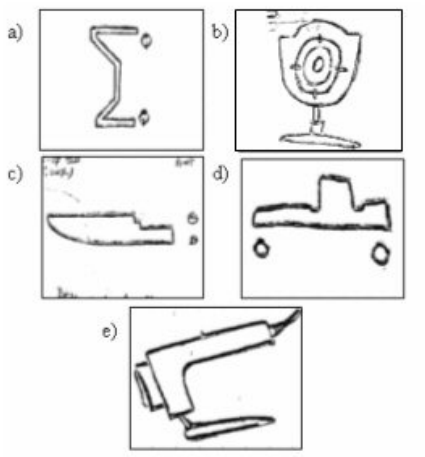

Figure 13: *Illustrates the results obtained after the shadow removal technique was applied to the results obtained after binarisation shown in Figure 12.*

method are illustrated in Figure 13. Here the shadow boundary present in Figure 12(c) has been successfully removed.

However, this method is limited to shadows which are not aligned to the line drawing since this would result in the deletion of both shadow boundary and line drawing pixels. Thus, this technique requires further studies and is subject to future work.

Analysis of the time and space complexity of the algorithm gives an indication of the complexity of the algorithm and hence its performance with respect to time and memory usage when used with various images of different sizes. If the size of the image under test is denoted as $R \times C$, and the image is made up of 2^n different grey-levels, where n represents the resolution of the image, the implementation of the algorithm requires a total of $410 + 5 \times R \times C$ in memory space. Furthermore, the computation of the pixel class would require a total of $83 + 2^{n+5}$ multiplications and $(2^n(80 + 3 \times R \times C) + 112)$ additions. These values however, assume a worst case situation in which the image has the maximum possible contrast.

7. Conclusions

The adaptive parameter evaluation introduced in this paper allows the binarisation of a low-level digital image without the requirements of any user-defined parameters. This allows accurate binarisation of images without causing undue restrictions on the user. The modifications carried out on Brensen's algorithm makes it superior to many thresholding techniques since it may classify correctly image of poor quality, having inhomogeneous paper background. Since the algorithm is free from user defined parameters, it may be successfully integrated within an automatic sketched line drawing interpretation system which would allow the integration of the paper based sketch with the CAD tools, thus enhancing design processes.

References

[AP00] ABLAMEYKO S., PRIDMORE T., *Machine Interpretation of Line Drawing Images*, 2000

[BAR04] BARTOLO A., *Binarisation and Thinning for Sketched Line Drawing Interpretation*, Undergraduate dissertation, Department of Microelectronics, University of Malta, 2003

[BM04] BULEN S., MEHMET S., *Image Thresholding Techniques - A Survey Over Categories* Journal of Electronic Imaging 13, 1 (Jan 2004) 146-165

[GW02] GONZALEZ R.C., WOODS R.E., *Digital Image Processing*, Prentice-Hall, 2nd edition 2002

[KZ93] KAMEL M. AND ZHAO A.,*Extraction of Binary Character/Graphics from Grey Level Document Images*, CVGIP: Graphical Models and Image Processing, Vol.55, No.3, pp.203-217, 1993

[PM99] PETROU M. AND BOSDOGIANNI P. *Image Processing, The Fundamentals*, Wiley, 1st Edition, 1999

[TJ98] TRIER O.D., JAIN A.K.: *Goal Directed Evaluation of Binarisation Methods*. Workshop on Performance versus Methodology in Computer

Vision 17, 3 (Aug. 1994), 209-217. (Proc. NSP/ARPA 98).

[YY00] YIBING Y, YANG H., *An Adaptive Logical Method for Binarisation of Degraded Document Images*, Pattern Recognition Society, 33 (2000) 787-807.

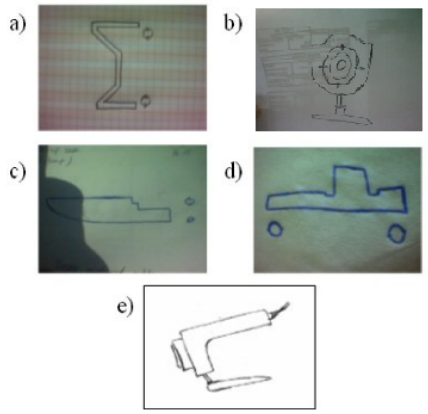

Figure 14: *Sample images digitized by means of digital cameras and scanner. Image (a) illustrates an image drawn on graph paper, Images (b), (c) and (e) illustrate images drawn on plain paper, whilst Image (d) shows an image drawn on a tissue paper. Images (a-d) were captured by means of a camera-phone whilst image (e) was captured with a scanner.*

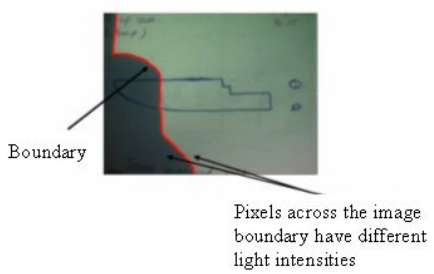

Figure 15: *Pixels located on either side of the image boundary, shown in red, have a large difference in their grey-level intensities.*