

# IMAGE BINARISATION USING THE EXTENDED KALMAN FILTER

Alexandra Bartolo, Tracey Cassar, Kenneth P. Camilleri  
*Department of Microelectronics*

Simon G. Fabri  
*Department of Electrical Power and Control Engineering*

Jonathan C. Borg  
*Department of Manufacturing Engineering*

*University of Malta*

Keywords: Feature Extraction, Image Processing, CAD.

Abstract: Form design is frequently carried out through paper sketches of the designer's mental model of an object. To improve the time it takes from solution concept to production it would therefore be beneficial if paper-based sketches can be automatically interpreted for importation into three-dimensional geometric computer aided design (CAD) systems. This however requires image pre-processing before initiating the automated interpretation of the drawing. This paper proposes a novel application of the Extended Kalman Filter to guide the binarisation process, thus achieving suitable and automatic classification between image foreground and background.

## 1 INTRODUCTION

Line drawing interpretation systems are used in engineering design as an interface between engineering or architectural drawings and computer-aided design (CAD) tools (Ablameyko S. and Pridmore T., 2000). Using similar principles, sketch recognition systems are being developed since it is acknowledged that designers can express their ideas more naturally by means of sketches (Roth-Koch S., 2000). Recent developments of such recognition systems have focused on online sketches obtained by means of PDAs or tablet PCs. Since such systems are online, the interpretation system has additional information about the drawing, for example, pen position and velocity. However, these systems lack the portability and flexibility of paper (Farrugia P. et al., 2004). In order to achieve this flexibility, the images must be pre-processed, such that line data may be extracted from the static image. Binarisation is one such process, which compensates for the noise introduced by the digitizing system. This work proposes the use of the Extended Kalman Filter (EKF) to guide the binarisation of images as a step towards the automation of the sketch recognition process which provides the necessary data to control rapid prototyping and manufacturing equipment. The proposed method improves

the binarisation of poor quality images, whilst reducing the complexity of the threshold selection process. This paper is divided as follows: Section 2 gives a brief review of binarisation techniques, Section 3 introduces the EKF and illustrates how this filter may be used to identify a suitable threshold for binarisation. This is followed by the results in Section 4 and conclusions in Section 5.

## 2 BINARISATION TECHNIQUES

Binarisation is the process by which grey levels within an image are classified as either foreground or background (Ablameyko S. and Pridmore T., 2000). The selection of a suitable binarisation technique is dependent on the type and quality of the images being used (Bulen S. and Mehmet S., 2004). Since the sketched line drawing interpretation system that is being developed is expected to process images from the field, it will typically process poor quality images, digitized using low-resolution devices, such as a camera-phone. This requires a more detailed and local analysis of the pixel distributions in order to select a suitable threshold for pixel classification. This section describes five established binarisation techniques originally proposed for line drawings and text images.

Palumbo and Guliano (Yang Y. and Yan H., 2000) use a fixed  $9 \times 9$  window to evaluate the class of each pixel within the image. The pixel value is determined according to the five  $3 \times 3$  local pixels within the  $9 \times 9$  window centered on the pixel in consideration. An initial threshold is used to determine the pixels which definitely belong to the background whilst the remaining pixels are classified using a different label assignment rule requiring the specification of three additional user defined parameters. Determining the values of these thresholds is not straightforward since they cannot be deduced from the image properties.

Niblack's algorithm (Bulen S. and Mehmet S., 2004) evaluates a threshold for each pixel within the image according to the mean and standard deviation of the pixels in a predetermined window  $W$  centered at the pixel in consideration. The pixel class is determined as follows:

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

where  $T(x, y) = \mu(x, y) + k \times \sigma(x, y)$ ,  $\mu(x, y)$  is the mean grey level of the pixels within the window,  $\sigma(x, y)$  is the standard deviation of these pixels and  $k$  is a user defined parameter. Niblack's algorithm requires two user defined parameters. The window size determines the number of pixels from which the mean  $\mu(x, y)$  and standard deviation  $\sigma(x, y)$  are evaluated. Thus, the window size should reflect the quality of the image background, on which prior knowledge is unavailable. The value of  $k$  is used to adjust the amount of print object boundary that is taken as part of the foreground, and is therefore dependent on the quality of the drawn line which is also an unknown quantity.

Eikvil's method (Trier O. D. and Jain A. K., 2000) is based on Otsu's thresholding technique (Gonzalez R. and Woods R. E., 2002), and makes use of two concentric windows  $L$  and  $S$ . 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 a parameter  $k$  which determines whether there is sufficient contrast between the two clusters. This indicates the effectiveness of Otsu's threshold on the selected region. Thus, if the difference between the two means is larger than  $k$ , 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  $S$ . This method requires the specification of three user defined parameters, of which  $S$  and  $L$  define window sizes, whilst  $k$  determines the thresholding method applied to the smaller window. The size of the smaller window  $S$  may be set to 3 which defines the smallest window centered on a pixel. However, the remaining parameters must be specified according to the particular image properties.

Kamel and Zhao's logical adaptive technique (Kamel M. and Zhao A., 1993) compares the grey level of the pixel in consideration with eight local averages in a pixel neighborhood of size  $(2SW + 1) \times (2SW + 1)$  where  $SW$  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 (Yang Y. and Yan H., 2000) proposed a method by which the two parameters  $SW$  and  $T$  are calculated adaptively. However, the adaptive evaluation of the parameter  $T$  requires another parameter  $\alpha$ . Yang and Yan (Yang Y. and Yan H., 2000) specify a range of values of  $\alpha$  for which suitable values of  $T$  may be obtained.

Brensen's method (Bulen S. and Mehmet S., 2004) may either classify a single pixel or a group of pixels simultaneously according to the contrast present within a selected window. The window's contrast is defined as  $C(x, y) = Z_{max} - Z_{min}$ , where  $Z_{max}$  and  $Z_{min}$  are the maximum and minimum grey levels within the window. If this contrast is smaller than a predefined value  $k$ , the pixels within the window belong to the same class, and the entire window may be assigned to a single class. However, if the contrast  $C$  is sufficiently large, then the pixels within that window belong to two different classes. Since the window has high contrast, a simple threshold based on the average gray level may be used to classify the pixels within this window. Thus, the threshold  $T$  is defined as  $T(x, y) = \frac{1}{2} \times (Z_{max} + Z_{min})$ . This method requires the specification of parameter  $k$  which may be evaluated adaptively using the method proposed in (Bartolo A. et al., 2004)

## 2.1 Drawbacks

Although the above methods may yield results of considerably good quality, the classification processes requires that a classifying criterion is evaluated for each pixel in the image. Furthermore, these algorithms require the specification of some parameter, such as a window size in order to evaluate the threshold. Although suggested values are specified for some algorithms, better results are obtained after fine-tuning the parameter to the characteristics of the image under test. Thus the performance of these methods is susceptible to image conditions. Methods for the adaptive evaluation for Brensen's and Kamel & Zhao's methods have been proposed, but these require considerable computational times, which slow down the product prototyping process. In this paper, we attempt to overcome these problems by modelling the sketch as a trajectory being tracked in time.

### 3 LINE TRACKING

A sketched drawing may be considered as a number of lines which interact at junctions or corners to create two-dimensional shapes. These lines may be considered as distinct entities, which can be described independently by some mathematical model. In this paper, two mathematical models which describe the position of a point on a line and its intensity are used to enable line tracking. Each line stroke is modelled as a trajectory propagating with a velocity  $v$  along a line subtending an angle  $\theta$  with the horizontal axis, thus the position of the trajectory at a time instant  $k + 1$  is given by Equation (2), where  $x_1$  and  $x_2$  are the vertical and horizontal coordinates on a plane.

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + v \times \begin{bmatrix} \sin \theta \\ \cos \theta \end{bmatrix} \quad (2)$$

Given a static, offline image, the velocity of propagation is irrelevant and may be assumed to be unity, thus making the  $(x_1, x_2)$  coordinates of the line dependent only on the direction  $\theta$ . The bilinear interpolation relationship given by Equation (3) (Gonzalez R. and Woods R. E., 2002), can be used to describe the intensity of a pixel in terms of its  $(x_1, x_2)$  coordinates,

$$z(k) = Ax_1(k) + Bx_2(k) + Cx_1(k)x_2(k) + D \quad (3)$$

where  $A$ ,  $B$ ,  $C$  and  $D$  are interpolation coefficients derived from the four neighbors of a pixel in the image with coordinates  $(x_1, x_2)$ .

In this way, each stroke in a given drawing may be considered as a process modelled by Equation (2). From the original image, measurements of pixel intensities, given by Equation (3), can be obtained. By considering the  $(x_1, x_2)$  coordinates of each line as the states of a dynamic system and the intensity values to be measurements obtained from noisy sensors, Kalman filter theory (Maybeck P. S., 1982) may be used to estimate the system states and hence the coordinates of points on each line.

Since the relation between pixel intensity and pixel position, given by Equation (3), is not linear, the Extended Kalman Filter (EKF) was adopted. The EKF linearizes the state estimation around the current estimate by using the partial derivatives of the process and measurement functions to compute estimates even when non-linearities are present. However, stability of the EKF assumes that the process is linear around the current state (Maybeck P. S., 1982).

#### 3.1 The Extended Kalman Filter

In general, the EKF addresses the problem of estimating the state  $\mathbf{x}$  of a discrete-time process modelled by

nonlinear state-space equations of the general form:

$$\mathbf{x}(k+1) = \mathbf{f}(k, \mathbf{u}(k), \mathbf{x}(k)) + \mathbf{w}(k) \quad (4)$$

$$\mathbf{z}(k) = \mathbf{h}(k, \mathbf{x}(k)) + \mathbf{v}(k) \quad (5)$$

where  $\mathbf{w}$  represents the model noise and  $\mathbf{v}$  the measurement noise, which are zero-mean, Gaussian noise sequences of covariance  $Q$  and  $R$  respectively,  $\mathbf{u}$  is a known input,  $\mathbf{z}$  is a measured output and  $\mathbf{f}$  and  $\mathbf{h}$  are general non-linear functions. In our line sketching application, the line generator model given by Equation (2) is cast into the state space form of Equation 4 to give Equation (6). Similarly, Equation (3) which represents the intensity model, is cast into the state space form of Equation (5) to give Equation (7).

$$\mathbf{x}(k+1) = \mathbf{x}(k) + \theta(k) + \mathbf{w}(k) \quad (6)$$

$$\mathbf{z}(k) = \mathbf{h}(k, \mathbf{x}(k)) + \mathbf{v}(k) \quad (7)$$

where the state vector

$$\mathbf{x} \equiv [x_{11}, x_{12}, \dots, x_{n1}, x_{n2}]^T$$

represents pen positions on  $n$  line strokes in the image,

$$\theta(k) \equiv [\sin(\theta_1), \cos(\theta_1), \dots, \sin(\theta_n), \cos(\theta_n)]^T$$

represents the orientation of the lines. The intensity of the trajectories at time  $k$ , is given by  $\mathbf{z}$ , which may be written as:

$$\begin{aligned} z_1(k) &= A_1x_{11}(k) + B_1x_{12}(k) + \\ &\quad C_1x_{11}(k)x_{12}(k) + D_1 + v_1(k) \\ z_2(k) &= A_2x_{21}(k) + B_2x_{22}(k) + \\ &\quad C_2x_{21}(k)x_{22}(k) + D_2 + v_2(k) \\ &\quad \vdots \\ z_n(k) &= A_nx_{n1}(k) + B_nx_{n2}(k) + \\ &\quad C_nx_{n1}(k)x_{n2}(k) + D_n + v_n(k) \end{aligned}$$

where  $A_i$ ,  $B_i$ ,  $C_i$  and  $D_i$  are known constants obtained from the bilinear interpolation, given by Equation (3).

Comparison between Equations (4) and (6) gives  $\mathbf{f}(k, \mathbf{u}(k), \mathbf{x}(k)) = \mathbf{x}(k) + \theta(k)$ , and  $\mathbf{h}(k, \mathbf{x}(k))$  is given by Equation (3). Process noise  $\mathbf{w}(k)$  models any deviations from the ideal line stroke, and measurement noise  $\mathbf{v}(k)$  corresponds to noise affecting the intensity measurements of the image.

Based on the state space model given by Equations (6) and (7), the EKF algorithm is used to find an estimate  $\hat{\mathbf{x}}(k|k)$  to the actual state  $\mathbf{x}(k)$ , based upon intensity measurements  $\mathbf{z}(k)$  as follows (Maybeck P. S., 1982):

The Kalman Gain is defined as:

$$\mathbf{K}(k) = \frac{\mathbf{P}(k|k-1)\nabla_{\mathbf{h}(\hat{\mathbf{x}}(k|k-1))}^T}{\nabla_{\mathbf{h}(\hat{\mathbf{x}}(k|k-1))}\mathbf{P}(k|k-1)\nabla_{\mathbf{h}(\hat{\mathbf{x}}(k|k-1))}^T + R} \quad (8)$$

where

$$\nabla_{\mathbf{h}(\mathbf{x})} \equiv \begin{pmatrix} \frac{\partial \mathbf{h}_1}{\partial x_{11}} & \frac{\partial \mathbf{h}_1}{\partial x_{12}} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \frac{\partial \mathbf{h}_n}{\partial x_{n1}} & \frac{\partial \mathbf{h}_n}{\partial x_{n2}} \end{pmatrix}$$

represents the rate of change in intensity in the vertical and horizontal directions for each line stroke.

The state estimate is obtained by:

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{K}(k) [\mathbf{z}(k) - \mathbf{h}(k, \hat{\mathbf{x}}(k|k-1))] \quad (9)$$

where  $\mathbf{z}(k)$  represents the true intensity measured from the image and  $\mathbf{h}(k, \hat{\mathbf{x}}(k|k-1))$  is given by Equation (3) evaluated at  $\mathbf{x}(k) = \hat{\mathbf{x}}(k|k-1)$ . The error Covariance  $\mathbf{P}(k|k)$  is given by:

$$\mathbf{P}(k|k) = [\mathbf{I} - \mathbf{K}(k)\nabla_{\mathbf{h}(\hat{\mathbf{x}}(k|k-1))}] \mathbf{P}(k|k-1) \quad (10)$$

where the covariance prediction is:

$$\mathbf{P}(k+1|k) = \mathbf{J}_{\mathbf{f}(\hat{\mathbf{x}}(k|k))} \mathbf{P}(k|k) \mathbf{J}_{\mathbf{f}(\hat{\mathbf{x}}(k|k))}^T + \mathbf{Q} \quad (11)$$

where  $\mathbf{J}_{\mathbf{f}(\mathbf{x})}$  is the Jacobian matrix  $\left[ \frac{\partial f_i}{\partial x_j} \right]$ , which in this case is equivalent to the identity matrix from Equation (6). The state estimate prediction at time  $k+1$  is:

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{f}(k, \mathbf{u}(k), \hat{\mathbf{x}}(k|k)) \quad (12)$$

The EKF Equations (8) to (12) will recursively compute the state estimate  $\hat{\mathbf{x}}(k|k)$  for each iterate  $k$ , given initial estimates  $\hat{\mathbf{x}}(0|-1) = \mathbf{x}_0$  and initial covariance  $\mathbf{P}(0|-1) = \mathbf{P}_0$  which represents the initial covariance of the error, reflecting the initial uncertainty of the estimates.

### 3.2 Application to binarisation

The EKF equations described above, give the possibility of locating points on a number of lines that form part of the drawn object or objects in an image. In this way, the EKF helps to discriminate between the image foreground and background by selecting those pixels which are located on lines and are therefore part of the image foreground. The intensity of the pixels along the tracked trajectory provides information about the grey level intensities of the pixels forming part of the line drawing. The mean grey level intensity  $\mu_t$ , and standard deviation  $\sigma_t$  of the tracked pixels can therefore be used to approximate the grey-level intensity of the sketched object, and hence guide the binarisation process. Since line pixels are darker than the background, the image pixels are classified by comparison to a threshold  $T = \mu_t + n \times \sigma_t$ , where  $n$  is a constant which defines the tolerance to grey-level variations along the tracked line. Pixels whose intensities are less than  $T$  may be classified as foreground line pixels whilst the remaining pixels may be classified as background pixels.

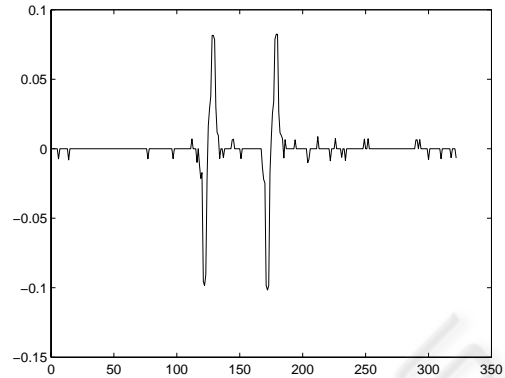


Figure 1: The derivative of a horizontal row in an image. Each positive - negative peak indicates the presence of a line.

### 3.3 Implementation of the EKF

The implementation of the EKF requires suitable starting points to initialize the line tracking process. The position of the lines in a static image are unknown, however, it may be assumed that part of the image will be located towards the center of the image. Thus, two scans along the horizontal and vertical centerlines of the image are performed. The derivative of the grey level intensities of pixels lying along this line is considered. The presence of a line is indicated by a negative and positive peak along the tracking direction, which correspond to the two edges of the line stroke as shown in Figure 1. Suitable starting points may be taken at the midpoints of two such peaks. Since a horizontal and a vertical scan are carried out, the EKF can be initialized with at least two starting points.

Since sketches are not necessarily built from straight lines, but may have curves or lines that exhibit a change in orientation, the sketched strokes are modelled as piece-wise linear segments. This implies that the line stroke may be built from a number of short straight line segments which may be represented by the state space model shown in Equation (6), thus requiring the evaluation of the line orientation  $\theta(k)$  at each iteration  $k$ . This may be obtained by using Sobel edge response (Gonzalez R. and Woods R. E., 2002), which gives the magnitude response and the orientation of the pixels within the image. For edge pixels, the magnitude response is highest and the pixel orientation corresponds to the orientation of the line. Thus, for each state  $\hat{\mathbf{x}}$ , the closest edge pixel pair are located and their orientation is used as an approximation for the line direction  $\theta(k)$ .

The state estimates given by Equations (9) require the evaluation of the intensity of the image at instant

*k*. This may be obtained by searching for the darkest gray level within a distance of one unit from the current pixel position in the direction of  $\theta + \Delta\theta$  where the  $\Delta\theta$  term is used to allow for deviations in the line direction. Since the line strokes are expected to be smooth, any line deviation from  $\theta$  is accommodated by seeking the darkest pixels in the cardinal directions that enclose  $\theta$ .

Since the EKF assumes that the model used is linear around the current state, large deviations from the line stroke would cause the filter to diverge. For this reason, it is required to terminate the tracking before the filter diverges. An indication that the filter is diverging may be obtained by comparing the gray level intensity of the tracked point with the grey level intensity of background pixels. A measure of the background intensity may be obtained by taking the mean gray-level  $m_s$  of a sample of pixels located at a distance  $d$  perpendicular to the line direction, where  $d$  should be greater than the stroke width of the sketched lines. A point located on an image line will have a gray level  $m_p$  which is less than the mean gray level  $m_s$  of the sample pixels. Thus, divergence is indicated when the mean gray level  $m_s$  of the sampled pixels is less than or equal to the gray level intensity  $m_p$  of the tracked point thus indicating that the point is no longer on the sketched line and has moved to a background region. This criterion also detects when the tracking point arrives at the end of a line and is therefore also used as a criterion to terminate the tracking process once the end of the line is reached.

The image digitization process introduces some degree of noise to the image such that adjacent pixels will have variations in their gray level intensity even though they belong to the same class. This will introduce errors in the evaluation of the EKF starting points since the derivative of an image row or column will also reflect these intensity variations. For this reason the image is low pass filtered using a  $3 \times 3$  mean filter (Gonzalez R. and Woods R. E., 2002). This will reduce the effect of the gray level variations in adjacent pixels making the transition between image class more prominent in the row and column derivatives.

## 4 TESTING AND RESULTS

The proposed algorithm was tested under various conditions and compared to the performance of the other established methods discussed in Section 2. This section shows the results obtained for a number of sample grey level images, whose gray levels are in the range [0 256]. The visual results obtained by Kamel and Zhao's method are also shown in order to allow a visual comparison. Kamel and Zhao's method has been chosen as this has the lowest number of user de-

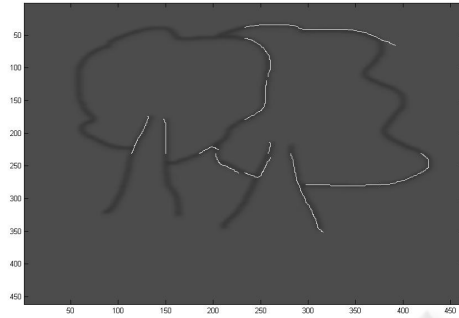


Figure 2: Illustrating the tracking paths generated by the EKF. White line segments indicate the tracked path, whilst the darker lines indicate the image line strokes. This example shows the tracking of 11 segments.

finned parameters and thus offers the highest degree of automation.

The ability of the EKF to track multiple lines was first tested with images having a low noise component as shown in Figure 2. This shows an example where the EKF tracks eleven segments, corresponding to seven starting points detected from a horizontal scan and four segments detected from a vertical scan of the image.

The algorithm was then tested under varying degrees of measurement noise. This was introduced by adding zero-mean Gaussian noise to the image. Figure 3 illustrates a ground truth image consisting of 4913 foreground pixels and 81651 background pixels. The corresponding noisy image is shown in Figure 3(b). The noise added has a standard-deviation of 36 grey-levels, which corresponds to a signal-to-noise ratio (SNR) of 12.8dB. The results obtained by the EKF algorithm and Kamel and Zhao's algorithm are shown in Figure 3(c) and 3(d) respectively. Further results are given in Table 1. These show that the results obtained by the EKF algorithm have lower

Table 1: Comparison of percentage pixel error of the EKF algorithm and Kamel and Zhao's algorithm under different noise conditions for the image shown in Figure 3(a). f→b indicates the percentage foreground misclassification and b→f the percentage background misclassification.

SNR(dB)	% pixel error			
	EKF		Kamel & Zhao	
	f → b	b → f	f → b	b → f
18.08	7.1	0.53	17.9	0.79
16.20	14	0.07	26	1.19
14.54	12.2	0.18	35.8	1.59
12.79	2.9	0.97	44.7	1.9

Table 2: Number of lines tracked and the sample points considered, given as a percentage of the total number of pixels in the image for the images shown in Figure 4

Image	Lines Tracked	% Points Tracked
4(a)	7	0.24
4(b)	4	0.27
4(c)	7	0.36

Table 3: Comparison of computational times with the algorithms discussed in Section 2 for the images shown in Figure 4

Image	Computational Time (s)		
	4(a)	4(b)	4(c)
EKF	57.3	26.3	55.9
Eikvil	75.2	73.7	75.4
Brensen	160	139.5	137.9
Kamel	78.6	78.8	70.9
Niblack	30.7	30.5	39.8
Palumbo	12.5	9.7	13

foreground misclassifications and background misclassifications in comparison to the results obtained by Kamel and Zhao's method. This indicates that the EKF algorithm gives a better performance than that of Kamel and Zhao under noisy conditions.

Figure 4 illustrates three images used to test the algorithm. These include multiple lines and curves, which illustrate that the line tracking process may effectively track such images. The number of lines tracked and the sample points taken is tabulated in Table 2.

This indicates that the thresholding decision is based on a small number of pixels, which however, correspond to pixels directly related to the image foreground. The binary result obtained for these images is illustrated in Figure 5. The results obtained may be compared with those illustrated in Figure 6, which are obtained by using Kamel and Zhao's algorithm after *manually* determining the most suitable values for  $\alpha$ . These results show that the proposed EKF algorithm gives results whose quality is comparable to those given by other binarisation techniques. Furthermore, Table 3 shows that the proposed binarisation process requires lower computational times than Brensen's and Kamel & Zhao's methods, for which adaptive parameter evaluation was applied. Although Niblack's, and Palumbo and Guiliano's methods show lower computational times, this does not include the time required to find suitable parameters for each image, because, these are not adaptive methods.

The algorithm was also tested on images captured by a cameraphone at a resolution of 96 dpi. A sam-

ple of these images is shown in Figure 7 where Figure 7(a) shows an image drawn on plain white paper, Figure 7(c) shows an image drawn on textured tissue paper whilst Figures 7(e) and 7(g) show images drawn on graph paper. Note that the cameraphone captures the light reflections on the image such that the resulting digital images display variable gray-level intensities along the background. This is particularly evident in the image corners, where the gray-level intensity of the background is comparable to that of the foreground. The performance of the EKF algorithm is comparable to that of the algorithms discussed in Section 2 but require smaller computational times. Figure 7 shows how the algorithm correctly identifies the foreground pixels. Of particular importance is the distinction made between the image foreground and the background grid in Figures 7(e) and 7(g), which allows the extraction of the object from interfering backgrounds. Some misclassification of the background pixels in the corner regions occurs for images in Figure 7(f) and 7(h). This is due to the fact that the image is being thresholded with a global threshold. The images shown in Figure 7(f) and 7(h) have some foreground pixels with a relatively high gray level intensity, resulting in a higher valued threshold, which will classify the darker background regions as foreground.

## 5 CONCLUSION

The proposed EKF binarization method has been shown to yield good quality results that are comparable to those obtained by existing methods. Further work is being carried out in order to apply the algorithm to local image regions, resulting in a number of local thresholds rather than a single global threshold. This will reduce the effect of global thresholding, thus further improving the results obtained.

The proposed method offers a higher degree of automation than the other binarization techniques discussed since no user defined parameters are required. This helps to improve the rapid prototyping process of the sketched line drawing, which is the main, long term objective of this work.

## ACKNOWLEDGEMENTS

This work has been mainly supported by Grant 73604 of the University of Malta.

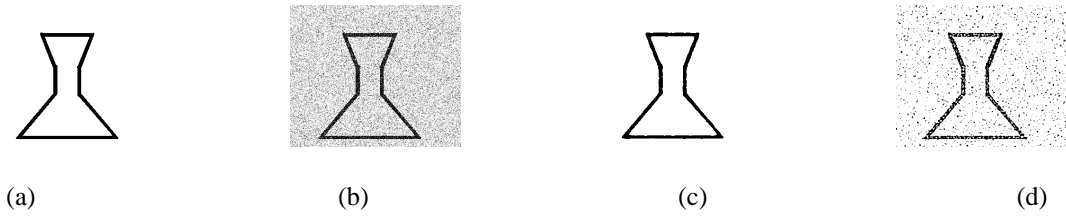


Figure 3: Illustrating the performance of the EKF binarisation under noise. Figure (a) shows a ground truth image and Figure (b) shows the corresponding image with added Gaussian noise resulting in an SNR of 15.5dB. Figures (c) and (d) show the binary result obtained by the EKF method and Kamel & Zhao's method respectively.

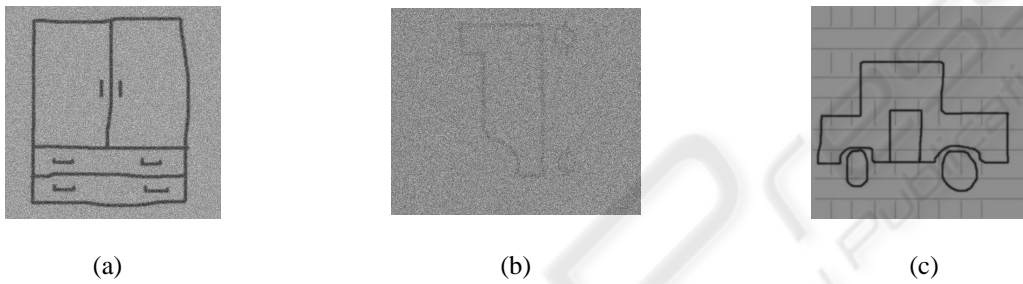


Figure 4: Sample test images. Images (a) and (b) illustrate image having a homogenous background with dynamic ranges of 160 and 80 respectively, whilst Image (c) is an example of an image having background artefacts.

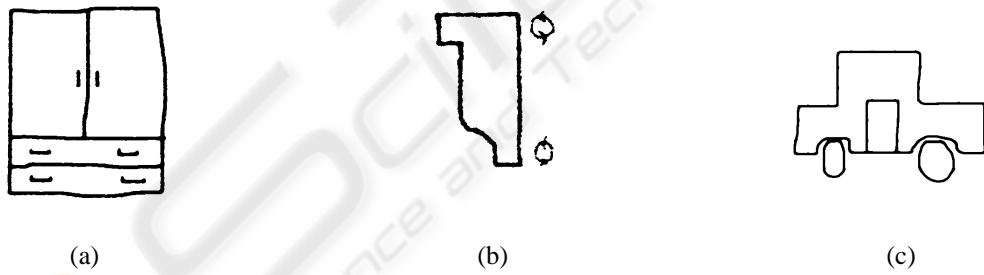


Figure 5: Results given by the EKF binarisation for images shown in Figure 4.

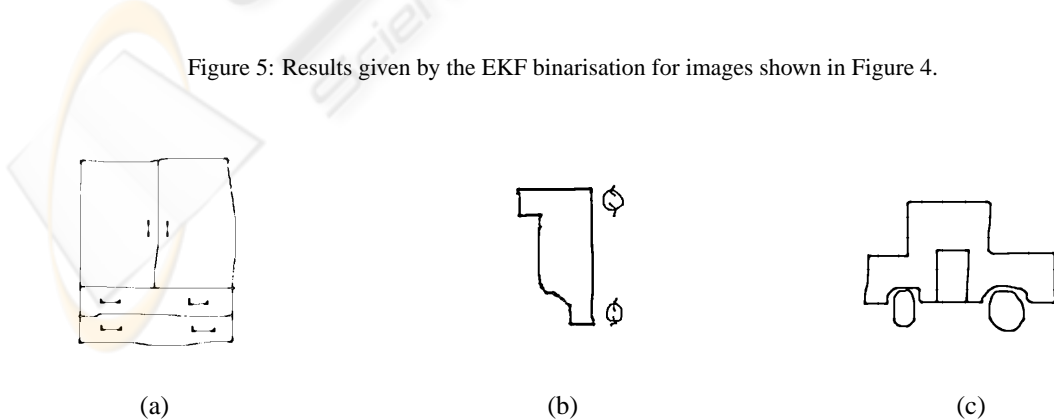


Figure 6: Results given by Kamel & Zhao's method for images shown in Figure 4, with  $\alpha$  equal to 0.4, 0.2, 0.1 respectively.

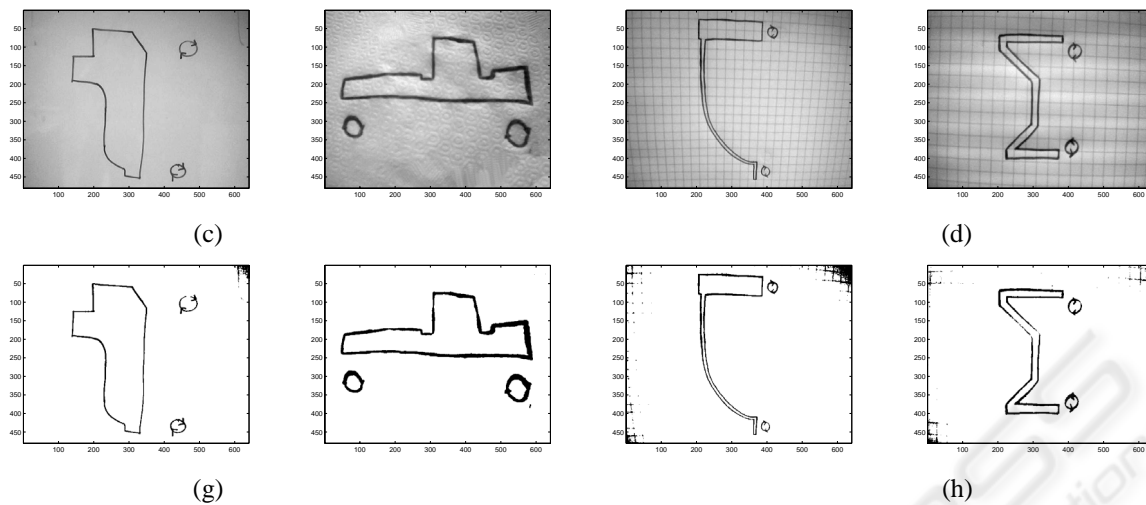


Figure 7: Results obtained by the EKF binarisation algorithm for a set of real test data captured with a cameraphone using a resolution of 96 dpi. Figures a), c), e) and g) show the original images whilst Figures b), d), f) and h) show the binary result.

## REFERENCES

- Ablameyko S. and Pridmore T. (2000). *Machine Interpretation of Line Drawing Images*. Springer-Verlag.
- Bartolo A., Camilleri K., Borg J., and Farrugia P. (2004). Adaptation of Brensen's Thresholding Algorithm for Sketched Line Drawings. *Eurographics Workshop on Sketch-Based Interfaces and Modeling*, pages 81–90.
- Bulen S. and Mehmet S. (2004). Image Thresholding Techniques - A Survey over Categories. *Journal of Electronic Imaging*, 13:146–165.
- Farrugia P., Borg J., Camilleri K., Spiteri C., and Bartolo A. (2004). A Cameraphone-Based Approach for the Generation of 3D Models from Paper Sketches. *Eurographics Workshop on Sketch-Based Interfaces and Modeling*, pages 34–42.
- Gonzalez R. and Woods R. E. (2002). *Digital Image Processing*. Prentice Hall.
- Kamel M. and Zhao A. (1993). Extraction of Binary Character/Graphics from Grey Level Document Images. *CVGIP: Graphical Models and Image Processing*, 55(3):203–217.
- Maybeck P. S. (1982). *Stochastic Models, Estimation and Control, Volume 2*. Academic Press.
- Roth-Koch S. (2000). Generating CAD Models from Sketches. *Proceedings of the IFIP WG5.2 Geometric Modeling: Fundamentals and Applications*, pages 207–219.
- Trier O. D. and Jain A. K. (2000). Goal directed evaluation of binarisation methods. *Workshop on Performance versus Methodology in Computer Vision*, 17(3):209–217.
- Yang Y. and Yan H. (2000). An Adaptive Logical Method for Binarisation of Degraded Document Images. *Journal of the Pattern Recognition Society*, 33:787–807.