Adaptive Colour Segmentation in Digital Video Images by means of Artificial Neural Networks and Morphological Operations

MARCO KRIPS and ANTON KUMMERT
Communication Theory, Department of Electrical and Information Engineering
University of Wuppertal
Rainer-Gruenter-Strasse 21, 42119 Wuppertal
GERMANY
krips@uni-wuppertal.de, http://wetnt7.elektro.uni-wuppertal.de/welcome

Abstract: - There are many industrial products that use digital video image processing for orientation control or object detection. Often object shape is used for classification. However, if detection task has to be performed very fast and there are many different candidate shapes for an object, other characteristic features have to be used. One of the most relevant object features can be colour. For example, the detection of human hands is a challenging problem, where a shape-based approach is unpromising, if detection has to be done in real time. The presented approach proves that a robust detection can be obtained by applying a neural network to the RGB values of digital video images and subsequent use of morphological operations. Furthermore, the claim for very small processing times is considered in the presented solution.

Key-Words: - Artificial Neural Networks, Digital Image Processing, Binary Morphological Operations

1 Introduction

Due to decreasing prices for (digital) cameras and powerful computers, digital video image processing is of growing interest. Range of applications enlarges rapidly, especially in the field of industrial applications, where image processing was too expensive in the past or has taken to much time for computing results. Today applications like inspection of printed circuit boards (correct placement of components) are standard. Furthermore, video image analysis is applied to tasks where correct positions or orientations of components must be checked. For most of these tasks a monochrome camera and the shape of objects is used for classification. Nevertheless, due to decreasing prices of colour charge coupling device (CCD) cameras, the characteristic feature colour can be used for classification as well.

This paper will show how artificial neural networks and morphological operations can be combined to classify not only one colour but a set of colours, namely colours of a human hand, in digital video images. The presented approach also takes into account the demand for very low processing time in order to be useful for industrial and even safety applications (safeguards), where real time image processing is not only desired but also needed.

2 Artificial Neural Networks

It is impossible to scope all types of artificial neural networks and it is not desired to cover all different learning algorithms for the different networks. Hence, we concentrate our discussion on supervised learning strategies. Therefore, networks like self-organizing maps ([1]) are not considered. However, it will be demonstrated by considering some exemplary chosen network architectures, how different neural networks can be used for classification tasks for digital image processing with respect to hand detection. Further information on artificial neural networks can be found in [2], [3], or [4].

We turn our attention on perceptron and back-propagation, but solvability of the discussed problem is not limited to these networks. Here, it should be referred to radial-basis function (RBF) networks, which are discussed in [5] for example.

2.1 Perceptron

The perceptron is the simplest form of a neural network. It is described in detail in [6] by Minsky and Papert. The model of such a neuron consists of a linear combiner followed by a hard limiter (Fig. 1).

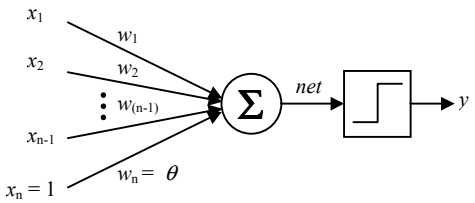


Fig. 1. Single-layer perceptron with fixed input for threshold

The output of this neuron can be described as

$$y = \begin{cases} 1 & \text{, if } net \geq \theta \\ 0 & \text{, else} \end{cases}, \qquad (1)$$

where net is defined by

$$net = \sum_{i=1}^{n} x_i w_i \quad . \tag{2}$$

The inputs of the perceptron are denoted as x_i and the synaptic weights by w_i. The threshold θ is applied as an additional constant input value that is multiplied by the threshold value.

Nevertheless, a single perceptron only can solve linearly separable problems. In other words, externally applied stimuli can be classified with respect to two classes. Although our problem obviously is not linearly separable, the result presented in this paper will demonstrate that the perceptron can be used anyhow.

2.2 Backpropagation

Another type of neural network discussed below is a multilayer feedforward network that is most common in many applications. The architecture of the neurons in these networks is similar to the perceptron with one exception. A sigmoid function, like hyperbolic tangent (tanh), is used instead of the hard limiter. Only for the output neuron the so-called logistic function is used. The logistic function is chosen in order to force a binary output. The standard training algorithm for such a type of neural network is backpropagation.

The original backpropagation learning algorithm, as presented in [3], has been modified in various aspects. In this context, the following algorithms can be mentioned, conjugate gradient algorithms [4] and the Broyden, Fletcher, Goldfarb, and Shanno (BFGS) update [7]. The advantage of all these variations relies on the fact that they are mostly more efficient than the standard backpropagation algorithm. In this work, we use the Marquardt-Levenberg algorithm [8] and RPROP [9] for neural network training.

2.2.1 Marquardt-Levenberg

The Marquardt-Levenberg algorithm (ML) is an approximation to Newton's method while backpropagation is a steepest descent algorithm. It minimizes the sum of squares of errors over all inputs. For Newton's method, the weight-updates are computed by

$$\Delta \mathbf{w} = -\left[\nabla^2 E(\mathbf{w})\right]^{-1} \nabla E(\mathbf{w}) \ , \tag{3}$$

where $\nabla^2 E(\mathbf{w})$ is the Hessian matrix and $\nabla E(\mathbf{w})$ is the gradient of the error function $E(\mathbf{w})$ that depends on the parameter vector $\mathbf{w}$ and has the form of a sum of squares. The Marquardt-Levenberg algorithm avoids computing the Hessian matrix. Instead, an approximation by means of the Jacobian matrix is used. The Jacobian matrix can be computed by a standard backpropagation technique that is much less complex than computing the Hessian matrix. The weight-updates are computed according to

$$\Delta \mathbf{w} = \left[J^T(\mathbf{w}) J(\mathbf{w}) + \mu \mathbf{I}\right]^{-1} J^T(\mathbf{w}) \mathbf{e}(\mathbf{w}) \ , \tag{4}$$

where $J(\mathbf{w})$ is the Jacobian matrix and $\mathbf{e}(\mathbf{w})$ is the corresponding error. The weight updating can be described as follows.

Whenever a step would result in an increased $E(\mathbf{w})$, the parameter μ is increased by a constant factor β and it is decreased by division by β, whenever a step reduces $E(\mathbf{w})$. So, if μ is large the algorithm becomes similar to steepest descent, whereas for small μ the algorithm becomes similar to Gauss-Newton.

2.2.2 Resilient propagation

Another algorithm we applied to the learning task is the resilient propagation (Rprop) algorithm ([9]). Rprop performs a local adaptation of the weight-updates $\Delta \mathbf{w}(t)$ according to the behaviour of the error-function. The Δw_{ij} are denoted by

$$\Delta w_{ij}(t) = \begin{cases} -\Delta_{ij}(t) & , \text{if } \frac{\partial E(t-1)}{\partial w_{ij}} \frac{\partial E(t)}{\partial w_{ij}} > 0 \ \wedge \ \frac{\partial E(t)}{\partial w_{ij}} > 0 \\ \Delta_{ij}(t) & , \text{if } \frac{\partial E(t-1)}{\partial w_{ij}} \frac{\partial E(t)}{\partial w_{ij}} > 0 \ \wedge \ \frac{\partial E(t)}{\partial w_{ij}} < 0 \\ -\Delta w_{ij}(t-1) & , \text{if } \frac{\partial E(t-1)}{\partial w_{ij}} \frac{\partial E(t)}{\partial w_{ij}} < 0 \\ -\operatorname{sgn}\left(\frac{\partial E(t)}{\partial w_{ij}}\right) \Delta_{ij}(t) & , \text{else} \end{cases} . \tag{5}$$

An individual update value Δ_{ij} is computed for each weight, which does not depend on the unforeseeable influence of the magnitude of the partial derivative but only depends on the behaviour of its sign. The update value Δ_{ij} is given by

$$\Delta_{ij}(t) = \begin{cases} \eta^+ \Delta_{ij}(t-1) & , \text{if } \frac{\partial E(t-1)}{\partial w_{ij}} \frac{\partial E(t)}{\partial w_{ij}} > 0 \\ \eta^- \Delta_{ij}(t-1) & , \text{if } \frac{\partial E(t-1)}{\partial w_{ij}} \frac{\partial E(t)}{\partial w_{ij}} < 0 \\ \Delta_{ij}(t-1) & , \text{else} \end{cases} . \tag{6}$$

The update value is increased by the factor η^+ whenever the partial derivative of the error function has the same sign for two successive iterations. The update value is decreased by the factor η^- whenever the partial derivative changes its sign. The update value remains the same, if the derivative is zero. When the weights are oscillating, the weight change will be reduced. If the derivative retains its sign, the update value itself is slightly increased in order to accelerate convergence in shallow regions.

3 Binary Mathematical Morphology

Mathematical morphology ([10]) can be described as shape-based image processing. In this work, only binary mathematical morphology is needed. A binary image can be treated as a 2D point set, where points belonging to an object are pixels with value equal to one and points belonging to the background are pixels with value equal to zero. Mode of operation can be understood by looking at two fundamental operations, dilation and erosion. More complex morphological operations such as opening or closing can be constituted from these two primary ones.

For graphical representation, we assume for the explanations in this section that the object is indicated by black blocks and the background by white blocks, while in the result section it is vice versa.

Binary dilation can be described as follows. If A, representing all object pixels, and SE, representing a structure element, are subsets of E^2, dilation is denoted by $A \oplus SE$ and is defined by

$$A \oplus SE = \left\{ \mathbf{c} \in E^2 \middle| \mathbf{c} = \mathbf{a} + \mathbf{se}, \mathbf{a} \in A \text{ and } \mathbf{se} \in SE \right\}. \quad (7)$$

Fig. 2 shows an example of dilation, where the set A of object elements is shown on the left-hand side, the structure element in the middle, and the result of dilation operation on the right-hand side respectively.

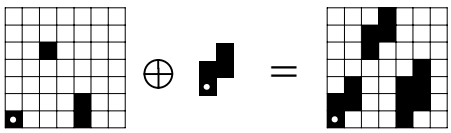


Fig. 2. Example for dilation on a binary image. On the left-hand side there is the set A representing all object elements, in the middle set SE representing the structure element, and on the right-hand side there is the result of dilation operation. The white dot specifies the origin.

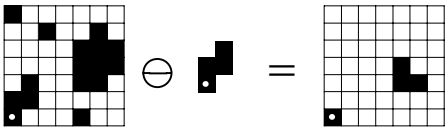


Fig. 3. Example for erosion on a binary image. On the left-hand side there is the set A representing all object elements, in the middle set SE representing the structure element, and on the right-hand side there is the result of erosion operation. The white dot specifies the origin.

The dual operator of dilation is erosion, which is denoted by $A \ominus SE$ and defined by

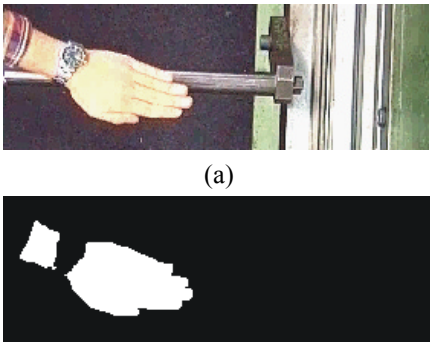
$$A \ominus SE = \left\{ \mathbf{c} \in E^2 \middle| \mathbf{c} + \mathbf{se} \in A \text{ for every } \mathbf{se} \in SE \right\}. \quad (8)$$

The duality of morphological operations is deduced from the existence of the set complement. Fig. 3 gives an example of dilation. The set A of object elements is shown on the left-hand side, in the middle the structure element is given, and on the right-hand side the result of erosion operation is presented.

Although dilation and erosion are dual operations, they are not inverse to each other. In particular, holes included in objects or indentations and projections cannot be reconstructed.

4 Training and Set-up

A binary reference output image had been manually created by means of graphical software tools since the learning task for perceptron and backpropagation is done by supervised learning algorithms. The reference output contains "1" where the corresponding input pixel belongs to a human hand and to the skin respectively. A "0" should be forced by the neural network for all other pixels (background) (see Fig. 4).



(a)

(b)

Fig. 4. Training pattern. On top (a) the training input and on bottom (b) the reference output created by means of graphical software are illustrated.

The real time aspect is the most important one in this approach. Therefore, a continuous data stream is processed rather than a full image. More precisely, the inputs of the first processing unit, the neural network, are the RGB values of single pixels i.e. the neighbourhood is not evaluated in this step. The representation of the pixels by RGB can be provided by the camera directly, thus no time-consuming space transformation is required. Furthermore, it is independent of different standards e.g. NTSC, PAL, or SECAM. The sufficiency of the RGB representation is also explained in [11]. Consequently, the artificial neural networks considered here have three input neurons and three inputs for the perceptron respectively.

The initial values for the weights and biases were randomly assigned to the multilayer feedforward networks as well as to the perceptron. Furthermore, the results of the neural networks are rounded to obtain binary output values.

After the classification by the ANN, some misclassifications still exist. Enhancing the detection result is

done by morphological operations that eliminate interfering misclassification effects as shown in the result section. At this step, the neighbourhood of pixels is considered as well. However, not the complete result image is needed to start processing but only the first n-1 lines of the result image and the first n pixel of line n, where n denotes the size of the n×n structure element. Thereby, low processing times can be achieved as the second processing unit can start without waiting until the first unit has finished processing of a whole frame. Output is a continuous data stream as well, from which a binary image can be reconstructed.

5 Results

Results can be discussed after different processing steps, the result after the classification by the ANN and the enhanced result after binary morphological operations.

5.1 Classification results

For the perceptron, 10,000 training cycles (epochs) were used, in contrast to this, training with Levenberg-Marquardt and Rprop was stopped after 1000 epochs. The mean square error (MSE) was used as performance function. Additionally, we define

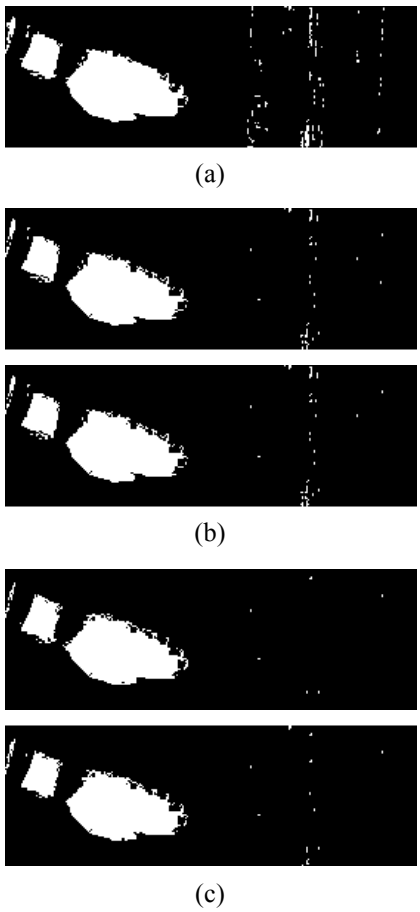
$$E_p = \sum_m \sum_n XOR(t_{m,n}, o_{m,n}), \qquad (9)$$

where $t_{m,n}$ is a pixel of the reference (teacher) image and $o_{m,n}$ is the rounded pixel value of neural network's output image. m denotes the number of lines and n the number of columns respectively. E_p gives the number of misclassified pixels with respect to the reference image. The numerical results for different neural networks are presented in Table 1.

		MSE	E_p	E_p [%]
Perceptron		0.01790	603	2.24
3/3/1	LM	0.01370	484	1.80
	Rprop	0.01402	482	1.79
3/3/4/1	LM	0.01247	449	1.67
	Rprop	0.01275	449	1.67
3/3/4/8/4/1	LM	0.01147	411	1.53
	Rprop	0.01264	442	1.64
3/3/9/18/9/1	LM	0.00974	366	1.36
	Rprop	0.01110	403	1.50

Table 1. MSE and E_p (absolute and relative value, based on total number of pixels in training image) for perceptron and four different multilayer feedforward networks, each trained with Levenberg-Marquardt (LM) and resilient propagation (Rprop) algorithm.

Obviously, an increased complexity of the neural network will lead to better results for MSE and reduced number of misclassified pixels E_p. Additionally, the slightly better performance of the Levenberg-Marquardt algorithm can be observed. However, the use of Rprop has the advantages that it needs less memory for computation and less time than LM for equal number of epochs.



(a)

(b)

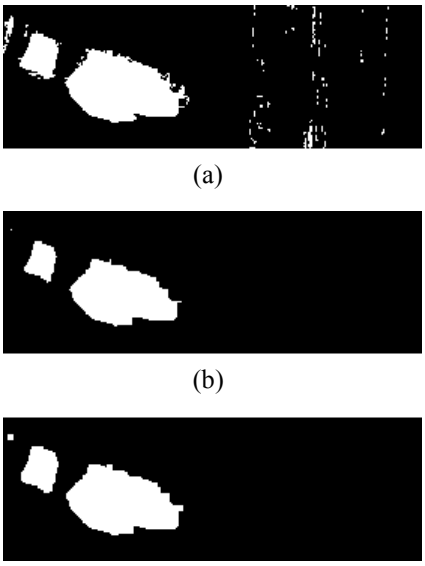
(c)

Fig. 5. Classification result of the perceptron (a), 3/3/1 neural network (b) trained with LM (upper) and Rprop (lower), and 3/3/9/18/9/1 neural network (c) trained with LM (upper) and Rprop (lower).

The visual assessment for perceptron, LM/Rprop 3/3/1, and LM/Rprop 3/3/9/18/9/1 results (Fig. 5) endorse these explanations. The number of misclassifications, especially in the background area, could be reduced by increasing complexity of the neural network. However, a higher complexity leads to larger processing times. Even if dedicated hardware is used, the processing time will increase for every additional layer added to the neural network. Consequently, another approach has to be used to enhance the results. Therefore, morphological operations are used to overcome this problem since these are less time consuming than an additional layer with neurons.

5.2 Result enhancement

The classification result can be improved by binary morphological operations. The computational effort associated with these algorithms is noticeable smaller than an additional layer with neurons. The perceptron, which has been the fastest of all networks considered here, needs one third more time to process the training image than the use of a combination of erosion and dilation applied to the binary output image of one of the neural networks needs. These timing results have been observed during simulation.



(a)

(b)

(c)

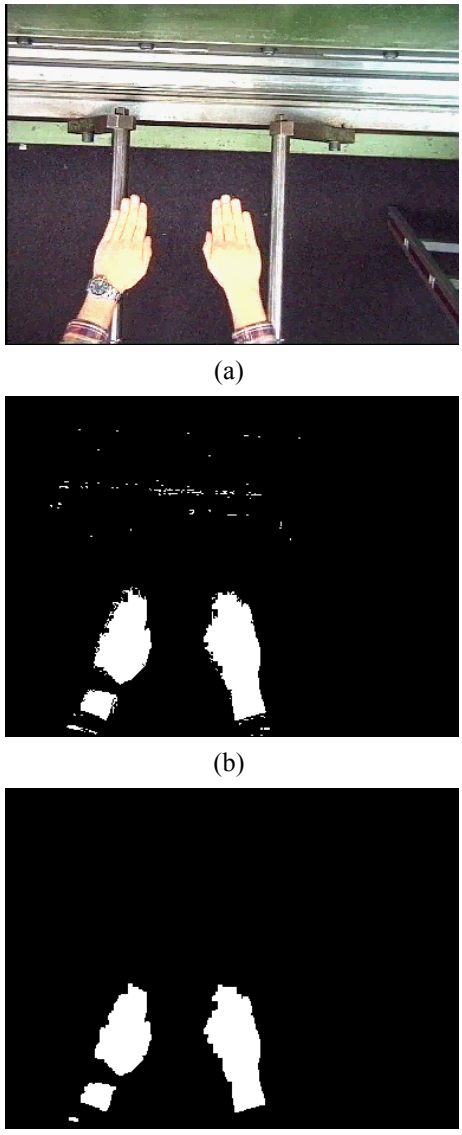
Fig. 6. Result enhancement of the perceptron's output (a) by applying erosion (b) and afterwards dilation operation (c). The structure element for both is a 4×4 neighbourhood.

Even the unsatisfying results of the perceptron can be improved by morphological operations (Fig. 6). At first, the erosion operation deletes the misclassifications. Here, size of structure elements depends on the size of these misclassified areas. Nevertheless, this erosion also leads to a shrinking of the detected hand. Thus, dilation with the same structure element is performed to regain approximately the original size of the hand. E_p is reduced from 603 down to 337 misclassified pixels. These remaining misclassified pixels are all located at the margins of objects as shown in Fig. 7.

Fig. 7. Classification errors of the enhanced output compared to the reference (teacher) image.

Covering of the hand's boundary areas can be obtained by using a larger dilation structure element. The precision of the shape will be slightly decreased, but the detected area will cover the hand completely. For better detection results, this boundary effect can be reduced, if the classification has been done with a higher precision. Therefore, more complex neural networks have to be considered, like a 3/3/1 architecture.

The other point, which has to be shown, is the generalization of the problem. Fig. 8 presents the classification results of a test image taken at an industrial facility.



(a)

(b)

(c)

Fig. 8. Result of LM 3/3/1 network with enhancement by erode and dilate operation with 3×3 structure element (test image (a), classification result (b), and the enhanced result after morphological operations (c))

A 3/3/1 LM network was used as classifier and result was enhanced by morphological erosion and dilation

with a 3×3 square structure element. All misclassifications in the surrounding of the machine have been deleted. Nearly all remaining objects belong to the human hands and arms. The neural network solely evaluates the colour information, which can be seen by looking at the classified part in the lower left of Fig. 8 (c). This object belongs to the shirtsleeve and has nearly the same colour as the skin of the person.

The presented video image analysis system could be realized on a today's up-to-date consumer computer (PC), but the real time capability mainly depends on the size of the neural network, because parallel processing as needed for larger neural networks with several layers is impossible. A hardware implementation of a suitable neural network with real time capability is presented in [12]. In addition, the hardware realisation of morphological operations is possible [13].

6 Conclusion

The combination of artificial neural networks and binary morphological operations leads to a fast colour detection system. The results reveal that a robust hand detection system can be obtained by this solution. Although the precision of classified objects is not perfect, the system can provide a full coverage of the object (hand) by adding some kind of safe zone.

Furthermore, processing time can be reduced considerably by using the neural network that has the smallest possible size. Removing of remaining misclassifications can be done by erosion and dilation. The binary morphological operations have the advantage that their processing time is less than that of an additional layer of neurons. Moreover, additional layers do not provide results without misclassifications in the background area. However, the latter can be obtained by binary morphology. Therefore, classified objects belong only to human hands and skin respectively, if no further major areas of objects with these set of colours exist in the background.

References:

[1] T. Kohonen, *Self-Organizing Maps*, Springer-Verlag, Berlin, 1995.

[2] S. Haykin, *Neural Networks - A Comprehensive Foundation*, Macmillan College Publishing Company, New York, 1994.

[3] D.E. Rumelhart, J.L. McClelland, *Parallel Distributed Processing - Exploration in the Microstructure of Cognition*, Vol.1: Foundations, MIT Press, Cambridge, Mass., 1986.

[4] M.T. Hagan, H.B. Demuth, M.H. Beale, *Neural Network Design*, PWS Publishing, Boston, Mass., 1996.

[5] T. Poggio, F. Girosi, *A theory of networks for approximation and learning*, A.I. Memo No. 1140, MIT, 1989.

[6] M. Minsky, S. Papert, *Perceptrons*, MIT Press, Cambridge, Mass., 1969.

[7] J.E. Dennis, R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice Hall, Englewood Cliffs, NJ, 1983.

[8] M.T. Hagan, M.B. Menhaj, "Training Feedforward Networks with the Marquardt Algorithm", *IEEE Trans. on Neural Networks*, Vol.5, No.6, 1994, pp. 989-993.

[9] M. Riedmiller, H. Braun, "A direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm", *Proc. of the International Conference on Neural Networks*, 1993, pp. 586-591.

[10] R.M. Haralick, S.R. Sternberg, X. Zhuang, "Image Analysis Using Mathematical Morphology", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, No.4, 1987, pp. 532-550.

[11] E. Littmann, H. Ritter, "Adaptive Color Segmentation - A Comparison of Neural and Statistical Methods", *IEEE Trans. on Neural Networks*, vol. 8, 1997, pp. 175-184.

[12] M. Krips, T. Lammert, and A. Kummert, "FPGA implementation of a neural network for real-time hand tracking system," *Proc. First IEEE International Workshop on Electronic Design, Test, and Applications*, Christchurch, New Zealand, 2002, pp. 313-317.

[13] J. Velten and A. Kummert, "FPGA-based implementation of variable sized structuring elements for 2D binary morphological operations," *Proc. First IEEE International Workshop on Electronic Design, Test, and Applications*, Christchurch, New Zealand, 2002, pp. 309-312.