A Modified 2D Chain Code Algorithm for Object Segmentation and Contour Tracing

Walid Shahab, Hazem Al-Otum, and Farouq Al-Ghoul
EE Department, Jordan University of Science & Technology, Jordan

Abstract: *In this paper, a modified algorithm for object segmentation of binary images is presented, and denoted as 2D modified chain code algorithm. The 2D modified chain code algorithm can be applied to color images after being binarized. The segmented object is used to derive the chain code in the image. The definition of the 2D- modified chain code algorithm is valid for shapes composed of triangular, rectangular, and hexagonal cells. The 2D modified chain code preserves information and allows computing geometric dimension. The results demonstrate that the 2D modified chain code algorithm could extract the coordinates of the shapes at lower computational cost when compared to the classical chain code. Here, a considerable improvement in accuracy (20.1-57.2%) over what is possible with the classical chain code has been achieved at the expense of slight increase in computational cost (10-20%).*

Keywords: *Image segmentation, contour detection, chain codes.*

Received September 11, 2007; accepted December 13, 2007

1. Introduction

Segmentation is a term used very commonly in computer vision and refers to several kinds of image decomposition/classification techniques. For example, segmentation is used in the extraction of contours and regions of an image or a scene [3], boundary detection [2], and voxel image analysis [11].

Generally, the first step in image analysis is to segment the image under consideration. Segmentation subdivides an image into its constituent parts or objects. The level to which this subdivision is carried depends on the problem being under processing. In general, autonomous segmentation (unsupervised segmentation) is one of the most difficult tasks in image processing. This task is the process that determines the eventual success or failure of the analysis. In fact, effective segmentation rarely fails to lead to a successful solution. Thus, considerable care should be taken to improve the probability of rugged segmentation. Numerous opportunities exist for application of segmentation in various stages from input image until segmentation and recognition. A general block diagram is illustrated in Figure 1.

One of the interesting techniques that are used in segmentation is the chain code technique. The first approach for representing digital curves using chain code was introduced by Freeman in [4]. Classical methods for processing chains are also introduced [5]. Chain code techniques are widely used to represent an object because they preserve information and allow considerable data reduction. A chain code approximates a curve with a sequence of directional vectors lying on a square grid [12] considered a detailed discussion of the properties of contour codes. Chen and Chen [1] proposed a simple recursive method for converting a chain code into a quadtree with a lookup table. However, this leads to an increase in the storage requirements. Nunes *et al* [6], developed a contour-based approach to binary shape coding using a multiple grid chain code which led to high computational cost. Truthe [10] proposed a synchronous chain code algorithm for picture languages. The algorithm is designed to decide the finiteness or infiniteness in polynomial time.

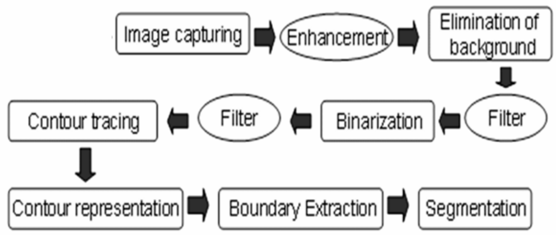


Figure 1. Segmentation stages.

Hierarchical search approaches the segmentation task by defining rectangles or search lines inside or along which individual digits or object contours were detected. This is the typical method of using regions of interests in image processing (search regions, test windows, areas of interest) to define the area within which image objects are to be segmented.

The success of feature-based inspection techniques depends on the quality of feature detection. Problems, such as edge detection and region extraction, to name the most important in 2D feature detection, belong to the mathematical class of inverse ill-posed problems. There exists no unique and stable transformation function that can build a specific description starting from an arbitrary observation. To overcome this problem, one has to reduce the number of acceptable solutions by introducing a priori knowledge of the problem space on the solution space. Thus, the detection process can be considered as decomposition into a sequence of sub-problems. This sequence is made of either well-posed problems or problems for which regularization methods exist.

In this paper, a modified chain code for shapes composed of finite number of cells is introduced. The definition of the modified chain code is based on extending the 1D into 2D chain code which is valid for shapes composed of triangular, rectangular, and hexagonal cells. The 2D-MCC has been applied for 2D object detection (failure detection) and results have shown that 2D-MCC has preserved 2D information and allowed computing of geometric dimension of objects under consideration. The organization of this paper is as follows. In section 2, preliminary information about the 1D chain code is given. In section 3, the proposed 2D-MCC is considered and the basic geometrical features of the proposed 2D-MCC Algorithm are discussed. Simulation results are discussed in section 4. Finally, concluding remarks are given in section 5.

2. Feature Extraction and the Chain Code

2.1. Introduction

The detection process consists of matching of the extracted features from the image under inspection with those of the predefined models. The detection process becomes very complex if the image to be detected is noisy and features could occur at random positions and orientations. This can be utilized by pixel gradients instead of by only pixel values, emphasizing the structure of the image instead of the texture. The gradient ∇f of an image *f(x, y)* at location *(x, y)* is given as:

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \tag{1}$$

It is well known, from vector analysis, that the gradient vector points in the direction of maximum rate of change of *f* at *(x,y)*. The magnitude $\|\nabla f\|$ of the gradient is generally referred to simply as the gradient and defined as:

$$\|\nabla f\| = \sqrt{G_x^2 + G_y^2} \tag{2}$$

This quantity equals the maximum rate of increase of *f (x, y)* per unit distance in the direction of ∇f. Common practice is to approximate the gradient with absolute values:

$$\nabla f \approx \left| G_x \right| + \left| G_y \right| \tag{3}$$

The decision regarding what features are to be considered depends on the nature of the object to be identified. Features provide data reduction while preserving the information required for the segmentation. Once an image has been clearly segmented into discrete objects of interest, the next step in the image analysis process is to measure the individual features of each object. Many features can be used to describe an object. Most of the procedures used for feature extraction involve edge detection, line tracing, and shape description techniques like the 1D chain code.

2.2. Basic Formulation of the Chain Code

The first approach for representing digital curves using chain code was introduced by Freeman in [4]. Freeman states that in general, a coding scheme for line structures must satisfy three objectives; it must faithfully preserve the information of interest, permit compact storage and be convenient for display, and facilitate any required processing. Chain codes are used to represent a boundary by a connected sequence of straight-line segments of specified length and direction. Typically, this direction is based on the 4 or 8 connectivity of the segments.

Let *R* represents the entire image regions. We may view segmentation as a process that partitions *R* into n-subregions: $R_1, R_2, \ldots, R_n$. Here, the following axioms must be satisfied**:**

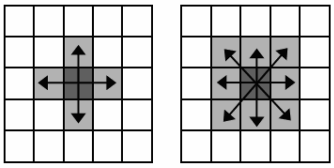
- $\bigcup_{i=1}^{n} R_i = R$
- R_i is a connected region, i = 1, 2, …, n,
- $R_i \cap R_j = ǿ$ for all i and j , i ≠ j
- $P(R_i)$ = True for i = 1, 2, …, n.
- $P(R_i \cup R_j)$ = False for i ≠ j.

where *P(Ri)* is a logical predicate over the points in set R_i and ǿ is the null set. Here:

- Axiom 1 indicates that every pixel must be in a region.
- Axiom 2 requires that points in a region must be connected.
- Axiom 3 indicates that the region must be disjoint, i.e., their intersection is null.
- Axiom 4 if all pixels in R_i have the same intensity.
- Axiom 5 indicates that region R_i and R_j are different in the sense of predicate *P*.

2.3. Pixel Connectedness

The contour of an object is commonly understood as a closed line running along the border of the object, i.e. consisting exclusively of border pixels. There are two common methods to define the neighborhood of a given image pixel in digital image processing: 4- and 8-connectedness, as depicted in Figure 2. The 4-connectedness allows only vertical and horizontal movements between adjacent pixels, whereas the 8-connectedness also uses diagonal connections.



(a) 4-connectedness. (b) 8-connectedness.

Figure 2. Different definitions of pixel neighborhoods.

2.4. Detecting Object Contours

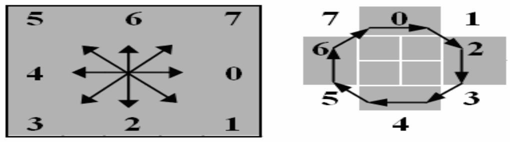
The principle of detecting objects by contour tracing is quite simple. Below, a short description is given based on Parkers algorithm [8]:

- The region of interest is searched for a first transition between background and object, i.e. a transition from (logical) white to black if dark objects are to be segmented, black to white for light objects.
- As soon as the transition has been found, the algorithm searches for the next neighbor of the same (logical) color with respect to the connectedness definition used. In this case, only border pixels are accepted as neighbors, i.e., pixels adjacent to at least one pixel of the other (logical) color in order to prevent the algorithm from swerving off into the interior of the object.
- Next, the entire contour is traced in this manner, either clockwise or counter-clockwise (the direction does not matter as long as one remains consistent).
- Contour tracing ends when the algorithm reaches its starting point, i.e., the pixel where the first transition has been found.

Contours generated using the above algorithm can be represented as a data structure in quite different ways. of these, the chain code has been generally accepted in digital image processing because it offers a number of algorithmic advantages. It is, for example, very simple to compute the perimeter and dimensions of an object from the chain code. It also facilitates smoothing the contour to lessen the influence of image noise.

The chain code, also called Freeman code [5], represents the contour as a sequence of directional information. Figure 3 shows how the direction of the connection between two contour points is encoded as a single number. The position of the starting point is stored separately. For better illustration, a flow chart of the traditional chain code [5] is depicted in Figure 4.

In this paper, a modified chain code is to be used to extract boundaries. The first step of the construction of the chain code is to extract the boundary of the image. Chains can represent the boundaries or contours of any discrete shape composed of regular cells. In this work, the length *l* of each side of the cells is set to one. These chains represent closed boundaries, and, thus, all chains are closed.



(a) the chain code. (b) a simple contour in chain code.

Figure 3. Contour representation by chain code.

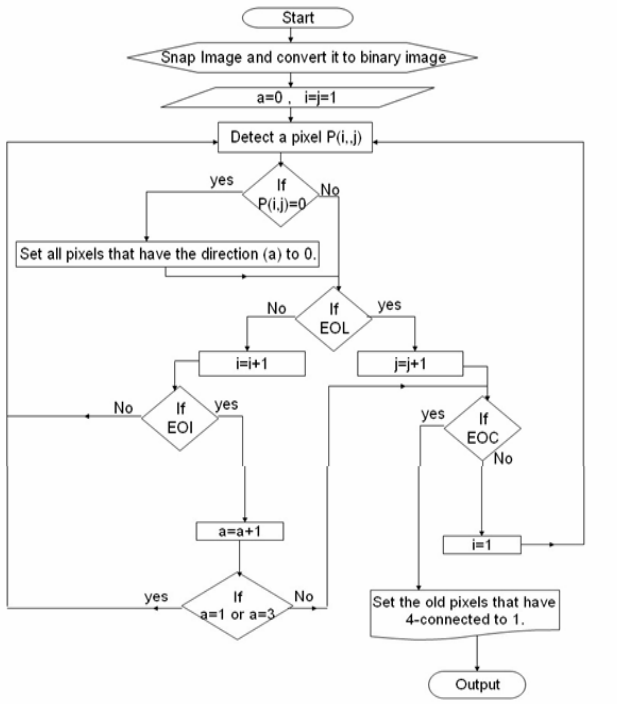


Figure 4. Flow chart of traditional chain code.

3. The Proposed 2D-MCC Algorithm

3.1. The Modified 2D Chain Code (2D-MCC)

Raji [9] developed an algorithm for determining the area of 2D objects by image analysis. This can be adapted in detection of leave type (as a form of sorting) to select the desired ones during harvesting. In this paper, a modified algorithm is considered to be implemented in finding the contour of a binary image, and, then to use this contour for objects segmentation. The proposed algorithm implements 4-connectivity rule but differs from 1D chain code by the fact that 2D-MCC is concerned of the position of each pixel. This is why the proposed algorithm is considered as two dimensional chain codes. Now, to find the contour of a binary image, i.e., to segment the object and to measure its dimensions, the proposed 2D-MCC takes place in the following steps.

Step 1: given the input image:

P ={p(i, j): i=1,2,…M & j = 1,2,…N }

Initialize the followings:

- I ={*I(ij)= 0* $\forall$ *i=1,2,..M & j=1,2,…N* }
- W ={*W(ij)= 0* $\forall$ *i=1,2,..M & j=1,2,…N* }
- T ={*T(ij)= 0* $\forall$ *i=1,2,..M & j=1,2,…N* }

Step 2: the input image is dealt in a row-based manner. For each pixel *p(i,j)* having a value *p(i,j)*= 0 (black), set the pixels that have the direction (0) in 4-connected to 0 as shown in Figure 5, i.e.,

$$I(i, j-1) = \begin{Bmatrix} 0, p(i,j) = 0 \\ 1, p(i,j) = 1 \end{Bmatrix} \tag{4}$$

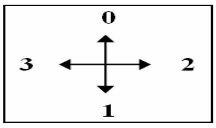


Figure 5. Directions of the neighbors.

Step 3: change the direction from 0 to 1 and set the pixels that have the direction (1) in 4-connected to (0).

$$I(i, j+1) = \begin{Bmatrix} 0, P(i,j) = 0 \\ 1, P(i,j) = 1 \end{Bmatrix} \tag{5}$$

Step 4: in the new image I, find the maximum and minimum values in the *x*-axis for each object *k*:

$$X_k = \begin{Bmatrix} n_{1k} = \min(i), \forall I(i,j) = 0 \\ n_{2k} = \max(i), \forall I(i,j) = 1 \end{Bmatrix} \tag{6}$$

where n_{1k}, n_{2k} represent left and right horizontal edges of the k^{th} object respectively.

Step 5: next, the image is dealt in a column-based manner, here, we skip the first and last column then repeat step 3 and 4 for directions 2 and 3, Figure 5. Here, for direction 2, we obtain:

$$I(i+1, j) = \begin{Bmatrix} 0, p(i,j) = 0 \\ 1, p(i,j) = 1 \end{Bmatrix} \tag{7}$$

and for direction 3:

$$I(i-1, j) = \begin{Bmatrix} 0, p(i,j) = 0 \\ 1, p(i,j) = 1 \end{Bmatrix} \tag{8}$$

Step 6: for the image *I*, find the maximum and minimum values in the *y*-axis for each object *k*:

$$Y_k = \begin{Bmatrix} m1k = min(j), \forall I(i,j) = 0 \\ m2k = max(j), \forall I(i,j) = 1 \end{Bmatrix} \tag{9}$$

where m_{1k}, m_{2k} represent top and bottom vertical edges of the k^{th} object respectively.

Step 7: detect pixels that have 8-connected zeros in the image *I*, then, use it in the already created zero image *W* as follows:

$$W(i,j) = \begin{Bmatrix} 1, & I(i-1, j-1) = \\ & = I(i-1, j) = I(i, j-1) = \\ & = I(i, j+1) = I(i+1, j-1) = \\ & = I(i+1, j) = I(i-1, j+1) = \\ & = P(i,j) = 0 \\ 0, & \mathit{Otherwise} \end{Bmatrix} \tag{10}$$

Step 8: perform 'and' logical operation between *P* and *W* at all pixel locations (*i, j*):

$$T = \{t(i,j) = p(i,j) \otimes w(i,j), \ \forall i = 1,2,\ldots,M \ \& \ j = 1,2,\ldots,N\} \tag{11}$$

Step 9: from step 4, 6 and 8, we obtain 4 points that represent corner points of the object. So, the object can be extracted from original image P. Once, the objects are extracted, their geometrical features can be easily obtained. Figure6 shows a simplified flow chart for the 2D-MCC algorithm.

A demonstrative example is shown in Figure 7, where the prescribed algorithm has been applied to various electronic components to obtain their contours. As well seen from Figure 7, the main advantage of this method when compared to other algorithms is that it always generates visually contiguous objects with closed border lines. The Basic geometrical features of the proposed 2D-MCC algorithm are discussed in the next section.

3.2. Basic Geometrical Features of the Modified 2D-MCC Algorithm

Under the rubric of basic geometrical features we subsume properties like coordinates, dimensions, areas, *etc.* Here, implementing such features are less algorithmically complex than shape-describing features, however, the distinction is naturally somewhat arbitrary.

3.2.1. Enclosing Rectangle

Every segmented object can be enclosed by a rectangle. In the simplest case this rectangle is oriented along the coordinate axes. The following basic features can be derived from:

- Position of the origin: the coordinates of the top left corner of the enclosing rectangle correspond to the x-coordinate of the leftmost point of the object and the y-coordinate of the topmost point of the object contour.
- Dimensions: the width or x-dimension of the rectangle is equivalent to the difference of the x-coordinates of the far right and far left points of the object contour. Correspondingly, the height or y-dimension is equivalent to the difference of the y-coordinates of the topmost and the bottommost points of the object contour.

- Ratio: this feature denotes ratio of height to width of the enclosing rectangle; this is already a very simple shape-descriptive feature.

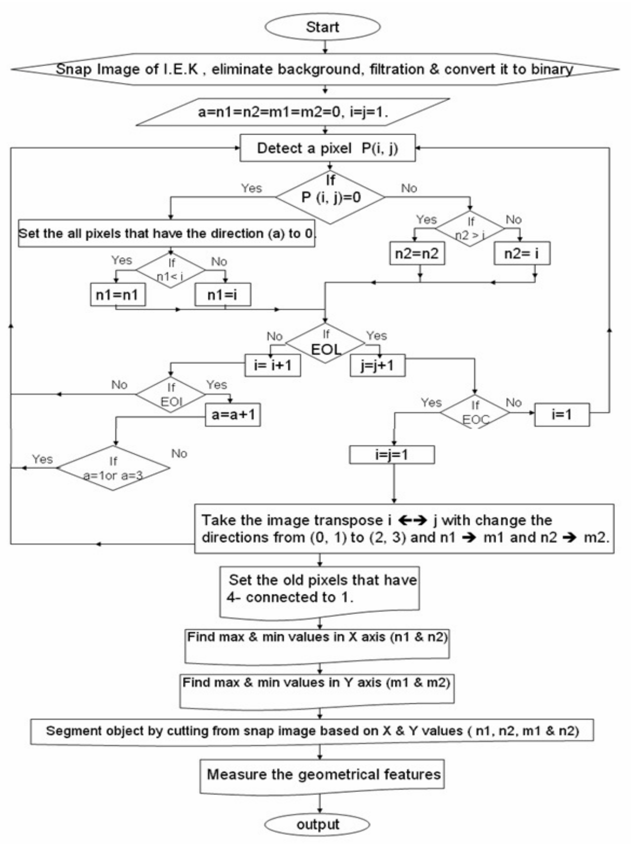
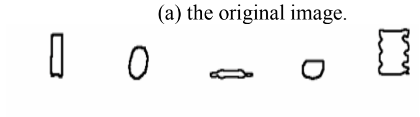


Figure 6. A simplified flow chart for the 2D-MCC algorithm.

(a) the original image.



(b) the contour of the image.

Figure 7. Electronic components.

If images are to be read line by line, they will therefore have to be sorted by x and y-coordinates. Usually the origin of the enclosing rectangle will be sufficient for this sorting operation and can be computed much faster than the objects center of gravity or may even be available from the segmentation algorithm anyway. It should be noted that for objects found by template matching, the enclosing rectangle is identical to the object border. Since an object segmented by template matching does not need to show a closed, visually identifiable contour, its limits can only be determined from the size of the template.

A more elaborate definition for the enclosing rectangle is the oriented enclosing rectangle that uses the direction of the principal axis of the object as coordinate axis. In this new coordinate system, the enclosing rectangle can be defined in the same way as above. Here, the rectangularity [8] is defined as:

$$R = A/A_{ser}. \tag{12}$$

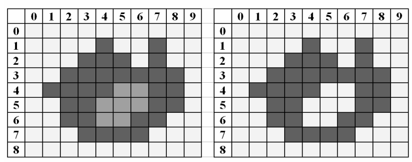
where A- the object area A_{ser} – the area of the smallest enclosing rectangle. Note that $R \leq 1$. $R=1$, for an exact rectangle, otherwise $R<1$. For an exact computation of this feature, the oriented enclosing rectangle has to be used, otherwise, a rotated exact rectangle would give a much smaller value than 1.

3.2.2. Area and Perimeter

In general, features like area and perimeter are meaningful only for objects segmented using chain code or other techniques since objects found by template matching do not necessarily correspond to a visually closed image structure. In the following, we will therefore always assume that objects are generated based on the proposed 2D-MCC and represented as contours or regions enclosed by contours.

The area of a segmented object is very easy to be computed. It is equivalent to the number of pixels inside the object contour. The area of a thresholded object can be defined in two different ways: 1) Filled area: the filled area encompasses all points inside the object contour including those that would not belong to the object according to the threshold , and, 2) Net area where the pixels, that do not belong to the object according to the threshold, are excluded.

Area computations are often performed using a run-length representation of the object. Each line of the object is represented as a series of segments, so called *runs*. All the pixels in each run belong to the object, i.e., runs do not contain non-object pixels. Its starting point and its length describe every run segment. The area of an object is then simply the sum of all run length. Figure 8 demonstrates the run-length encoding of an object featuring an indentation as well as a hole. After each line the corresponding segments are written out with their starting points in parentheses followed by the length of the segment.



(a) Filled area=38. (b) Net area=31.

Figure 8. Area computation based on run length encoding.

The perimeter of a segmented object can be computed directly from the 2D-MCC result. A caution has to be applied, though, because one cannot simply count the pixels of the contour. The following notes must be taken into account:

- The true contour runs neither along the outer edges of the contour pixels nor along their inner edges. Instead, it is located at some position in between, which depends on the combined effects of spatial discretization and gray level quantization, and is usually assumed to be in the center of the pixels.

- The distance between the center points of two pixels depends on the angle of the contour at that point, as shown in Figure 9. Here: 1) a vertical or horizontal connection between two pixels contributes a length of 1.0 to the perimeter, and, 2) diagonal connections contribute a length of $\sqrt{2} = 1.414$ to the perimeter.

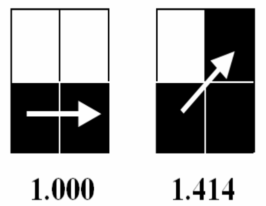


Figure 9. Contributions of contour segments to the perimeter.

Evidently, it is very simple to compute the perimeter from the 2D-MCC. Also, all diagonal contour pieces are encoded by odd values of the chain code, and, all vertical and horizontal segments by even values. The perimeter of the approximate circle in Figure10 is then given by: P = 8 * 1.0 + 8 * 1.414 = 19.312. An exact circle running through the centers of the contour points would have a radius of three pixels, resulting in a perimeter: P= 6π=18.85 pixels. Despite the small number of pixels, the deviation is only around 2.5 %. Nevertheless, perimeter values have to be used with caution, since; they depend strongly on the image resolution. With increasing magnification, more and more irregularities will become apparent in the contour, increasing the perimeter.

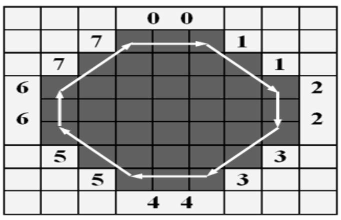
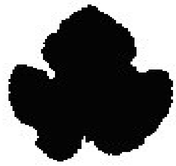


Figure 10. Perimeter computation on a circle after spatial discretization.

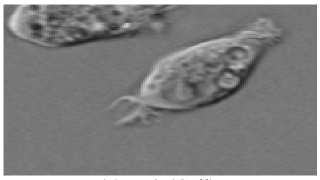
4. Simulations and Results

In this section, the proposed 2D-MCC algorithm will be tested using different test images. A data base of various image classes has been used. Here, images were: colored grayscaled and binary, examples of which are shown in Figure 11. These images are given the abbreviations Im1, Im2, and, Im3.

(a) Im1 (Electronic kit).



(b) Im2 (Tree).



(c) Im3 (Cell).

Figure 11. Sample test images.

A crucial issue is image binarization. Here, when color electronic kits, or even any color image, is to be binarized, some objects may disappear and other unimportant objects may be remained. To tackle this issue, three techniques were used:

- Thresholding: direct level thresholding is implemented to the color images after being converted to their grayscale version.
- Small Element: a small region of the background in the color image has been selected and resized to have the same size as the input image, then, the produced image is subtracted from the original color image.
- Subtraction: an effective method that is very suitable to electronic kits is that: an image of the Printed Circuit Board (PCB) has been captured before the planting process, then, the captured image (before planting) is subtracted from the PCB board (after planting).

The prescribed techniques were tested. Here, the third method, i.e., the subtraction, was only applied to color PCB images. For comparison purposes, an example of applying the three techniques to the test image Im1 is shown in Figure 12. As well seen, the subtraction technique gives the best accuracy ratio when compared to the direct thresholding as well as to the small element techniques.

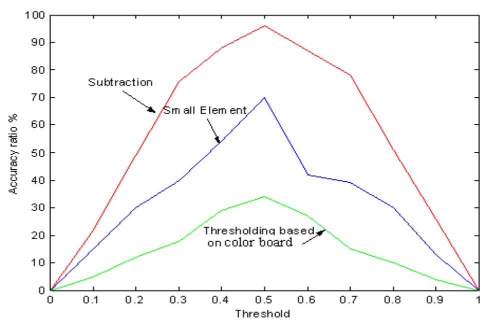


Figure 12. Results of implementing the three prescribed techniques of binarization with the test image Im1.

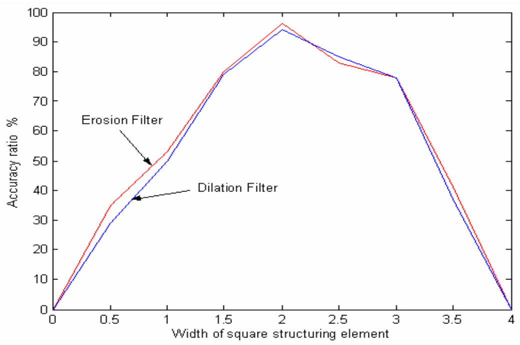


Figure 13. Results of applying morphological filter to the binarized Im1 with varying SE width.

A detailed study of the resulting binary images has shown that, as usually expected, the binary objects contain holes and noise-like pixels distributed randomly in the background. To overcome this problem, morphological filters can be implemented and play a crucial role in extracting boundaries [7]. In dilation or erosion operations, the boundaries are affected and some boundaries between two elements may be overlapped. Here, an important factor is the selection of the Structuring Element (SE) to obtain maximum accuracy ratio.

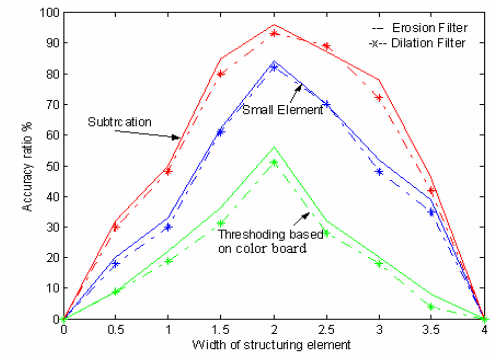


Figure 14. Results of implementing binarization to Im1 followed by erosion /dilation morphological filtration.

Figure 13 shows an example of the simulation results for Im1. Here it can be seen that the best value is around 2 pixels width. Also, Figure 14 depicts results of applying the three binarization techniques to the color image Im1, followed by the morphological filters (dilation or erosion). Again, we observe that the best value for the SE to obtain maximum accuracy ratio is 2 pixels width, and, the subtraction technique still the best to be implemented with electronic kit images as shown in Figure 12. Here, the SE size and type can be set based on the application under consideration. Analyzing Figures 12 and 14, the following notes can be remarked:

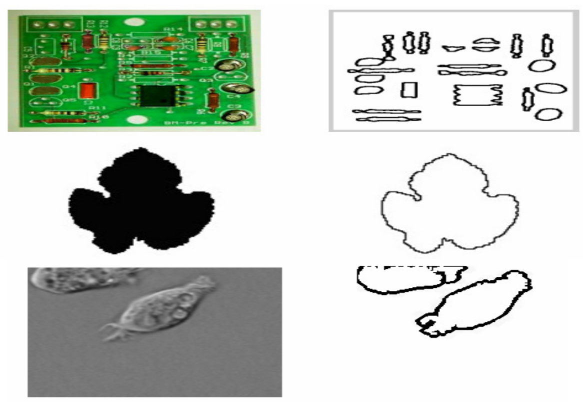
- The optimal range for thresholding, in the sense of the accuracy ratio, can be in the range [0.45– 0.55],
- The morphological erosion and dilation are very comparable, and, thus can be used equivalently. In fact, an open-close filter can be also used; however, a slight improvement was achieved at the expense of higher computation cost. This is why simple erosion/dilation operations were implemented.
- The selection of the SE size affects the accuracy ratio. In this work, SE width was set to 2. This will not lead to remove small components (in case of erosion), and, will not increase the noise level (in case of dilation).

Next, the resulting image is applied to the proposed 2D-MCC algorithm. Demonstrative examples are shown in Figure 15 (for the test images in Figure 11).

We now turn to the computational cost of the 2D-MCC algorithm. Let p be the number of subwindows in the image, n be the average size of a subpattern window (assuming a square window). Let t and s be the average number of window side translations and the average number of maximum window shifts, respectively. Assuming that r % of the image is blank and the rest of it has a pattern, thus:

- The amount of blank region processing cost $=4rpn$.
- The nonblank region processing cost $=4(1-r)pnts$, where the term $4n$ is the perimeter of a window.
- The percentage of image processed = $[(4rpn+a(1-r)pnts)/pn^2 x100=(4r+4(1-r)t)/n]$%. As t and s are usually less than 1 and very small compared to n, they can be dropped from the above equation, and the equation becomes: $[(4r + 4 (1-r))/n=4/n]$ %.

Typically, n>4; hence, the percentage of image processed by the proposed 2D-MCC algorithm is always less than 100%. For better illustration, Table 1 provides a comparison between the conventional Chain Code and the proposed 2D-MCC algorithm in the accuracy sense as well as the testing time for the selected test images Im1, Im2 and Im3. Here, the computational cost has been normalized by the maximum computational time for all cases, which was for the color image Im1. It is clear from Table 1 that there is a considerable improvement in accuracy (57.2%, 20.1% and 30.3 % for Im1, Im2, and Im3, respectively) at the expense of slight increase in computational cost (10-20%).



(a) The original image. (b) the contour image by 2D- MCC.

Figure 15. Comparision study.

Table 1. Comparative study.

Test Image	Method	Accuracy	Test Time
Im1	1-D Chain code	40.1 %	88.7 %
Im1	2-D Chain code	97.3 %	100 %
Im2	1-D Chain code	78.1%	55.2%
Im2	2-D Chain code	98.2%	65.2%
Im3	1-D Chain code	67.6%	65.2%
Im3	2-D Chain code	97.9%	73.3%

5. Conclusions

In this paper, a modified algorithm for object segmentation of binary images is presented, and denoted as 2D-MCC algorithm. The 2D-MCC algorithm can be applied to any binary image with or without holes. The segmented object is used to derive the chain code in the image. The definition of the 2D-MCC is valid for shapes composed of triangular, rectangular, and hexagonal cells. The 2D-MCC preserves information and allows computing geometric dimensions. The results demonstrate that the 2D-MCC algorithm could extract the coordinates of the shapes. Also, a considerable improvement in accuracy has been achieved (20-57%) when implementing the proposed 2D-MCC in comparison with the 1D chain code, but at the expense of slight increase in computational cost (10-20%).

References

[1] Chen Z. and Chen I., "A Simple Recursive Method for Converting a Chain Code Introduction a Quadtree with a Lookup Table," *Computer Journal of Image Vision and Computing*, vol. 19, no. 7, pp. 413-426, 2001.

[2] Ercal F., Moganti M., Stoecker W., and Moss R., "Detection of Skin Tumor Boundaries in Color Images," *Computer Journal of IEEE Transactions on Medical Imaging,* vol. 12, no. 3, pp. 132-141, 1993.

[3] Falah R., Bolon P., and Cocquerez J., "A Region Region and Region Edge Cooperative Approach to Image Segmentation," *IEEE International Conference on Image Process*, pp. 470-474, USA, 1994.

[4] Freeman H., "On the Encoding of Arbitrary Geometric Configurations," *in Proceedings of IRE Translation Electron Computer*, pp. 260-268, New York, 1961.

[5] Freeman H., "Computer Processing of Line Drawing Images," *in Proceedings of ACM Computer Surveys*, pp. 57-97, New York, 1974.

[6] Nunes P., Marqués F., Pereira F., and Gasull A., "A Contour-Based Approach to Binary Shape Coding Using a Multiple Grid Chain Code," *in Proceedings of Image Communication Special Issue on Shape Coding*, pp. 585-599, Germany, 2000.

[7] Oktem R, "Bar Code Localization in Wavelet Domain by Using Binary Morphology," *in Proceedings of the IEEE*, pp. 499-501, 2004.

[8] Parker R. and Wiley J., *Practical Computer Vision Using C*, Prentice Hall, New York, 1994.

[9] Raji O., "Discrete Element Modeling of the Deformation of Bulk Agricultural Particulates," *Computer Journal of Modeling, Design and Management of Engineering Systems*, vol. 2, no. 1, pp. 1-13, 2003.

[10] Truthe B., "On the Finiteness of Picture Languages of Synchronous Deterministic Chain Code Picture Systems," *Acta Cybernetica Archive*, vol. 7, no. 1, pp. 729-736, 2005.

[11] Vincken K., Koster A., and Viergever M., "Probabilistic Segmentation of Partial Volume Voxels," *Pattern Recognition Letters*, pp. 477-484, 1994.

[12] Wilson G., "Properties of Contour Codes," *in Proceedings of IEE Visual Image and Signal Processing*, pp. 145-149, 1997.

Walid Shahab received his BSc in electrical and electronic engineering in 1977, MSc in electrical and electronic engineering in 1978, and PhD in electrical and electronic engineering entitled "Surface Acoustic Wave Devices and Applications" in 1981 from the University of Science and Technique, France.

Hazem Al-Otum received his BS and MS in radio engineering (with Honors), and his PhD degree in digital communications and image processing from Graduate Theological Union.

Farouq Al-Ghoul received the BS and MS degrees in electrical engineering from Jordan University of Science and Technology, Jordan in 2002 and 2005, respectively.