Automatic Extraction of Graph-Like Structures from Binary Images

Marcin Iwanowski

Institute of Control and Industrial Electronics, Warsaw University of Technology
ul.Koszykowa 75, 00-662 Warszawa, Poland
iwanowski@isep.pw.edu.pl

Abstract. The paper describes a method for the analysis of the content of a binary image in order to find its structure. The class of images it deals with consists of images showing a groups of objects connected one to another forming a graph-like structure. Proposed method extracts automatically this structure from image bitmap and produces graph adjacency matrix describing it. The method is based on morphological image processing, skeletonization and labelling.

1 Introduction

The paper describes a method for the analysis of the content of a binary image in order to find its structure. The class of images it deals with consists of images showing at its foreground groups of objects connected one to another forming a graph-like structure. Described method extract automatically this structure from image bitmap and produces a matrix containing connections between all the objects shown on the input image - graph adjacency matrix. An example is shown in Fig. 1 - binary image shown on the left-hand side is characterized by graph-like structure of the foreground. This image consists of thicker and more compact parts which will be referred to as objects and thinner, elongated - further referred to connections. Such structure of the foreground resembles the graph where objects are its nodes, while connections are the edges of this graph. The graph describing the content of above mentioned image is shown on the right-hand side of the same picture. The proposed method is based on morphological image processing [3,9] and skeletonization [6,8,10]. In the first step binary objects presented on the image are classified into one of two groups: objects and connections. Classification of the image content based on spatial characteristics of image fragments follows an idea of morphological classification described in [2,1]. In the classification step, the morphological approach is used, which allows pixels' classifying based on theirs positions within the image. Next, the joints of connections to objects are detected. Finally the description of each joint is extracted based on labelings. This description consists of label assigned to joint, label of object and label of connection. The set of all above descriptions is stored in a temporary vectors which describes the graph structure present on the input image. Based on these vectors, the final product is computed - the

L. Bolc, J.L. Kulikowski, and K. Wojciechowski (Eds.): ICCVG 2008, LNCS 5337, pp. 461–468, 2009.

adjacency matrix which is of size $n_O \times n_O$ where n_O stands for the number of graph nodes (vertices, objects). In this matrix, both rows and columns refer to graph nodes, each element of this matrix has value equal 1 iff nodes related to both indices of this element are connected, and 0 otherwise. An example of such matrix describing the image from Fig. 1 is shown in Table 1. In the paper, an extended adjacency matrix is computed. Thanks to this extension, each element of this matrix contains the number of connections between every two nodes.

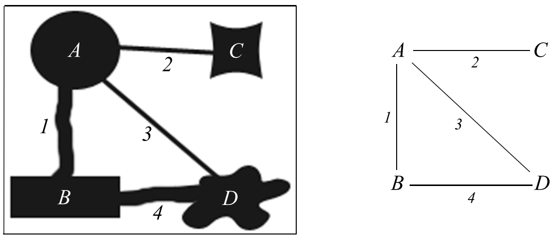


Fig. 1. Graph-like image content and related graph

Table 1. Adjacency matrix of graph shown in Fig. 1

	A	B	C	D
A	0	1	1	1
B	1	0	0	1
C	1	0	0	0
D	1	1	0	0

The paper is organized as follows. Section 2 introduces all the necessary tools used in the proposed method. Section 3 describes the method itself. Finally, section 4 concludes the paper.

2 Tools Used

2.1 Base Morphological Operators

Two base morphological operators are erosion and dilation [9,3], which - in the binary case - are defined as, respectively:

$$X \ominus B = \cap_{b \in B} X_b \; ; \; X \oplus B = \cup_{b \in B} X_b, \tag{1}$$

where X is a input image, B - is a structuring element and $X_b = \{x + b, x \in X\}$.

Structuring element B describes pixel's neighborhood. The simplest neighborhood in 2D case it is often *4-* or *8-connected elementary structuring element* and consists of a central pixel plus 4 (horizontal and vertical, $B = \mathcal{N}_4$) or 8 (horizontal, vertical and diagonal, $B = \mathcal{N}_8$) closest neighbors. Structuring element may also consist of neighboring pixels covering wider pixel neighborhood, of a given radius.

By combining erosion and dilation two base morphological filters [3,9,5] are defined: *opening* (erosion followed by dilation) and *closing* (dilation followed by erosion).

In the current study, opening operator is used to detect objects referring to graph nodes.

2.2 Anchored Homotopic Skeletonization

The second class of image content which is necessary to detect graph structure are connections between objects (nodes) which refer to graph edges. In order to perform this task, the homotopic thinning approach is used. This way of performing skeletonization is based on the notion of simple (or deletable) pixel [10,8] i.e. such a pixel which can be removed without changing the homotopy of the binary image.

A pixel p belonging to the image X is simple if and only if it satisfies the following three conditions:

1. $\mathcal{N}_{\mathcal{G}}(p) \cap X \neq \emptyset$,
2. $\mathcal{N}_{\mathcal{G}'}(p) \cap X^C \neq \emptyset$,
3. $\exists S \in \mathcal{CC}_{\mathcal{G}}(\mathcal{N}_8(p) \cap X)$ such that $\mathcal{N}_{\mathcal{G}}(p) \cap X \subseteq S$,

where X^C stands for the complement of image X. $\mathcal{N}_{\mathcal{G}}$ and $\mathcal{N}_{\mathcal{G}'}$ represents the closest neighborhood of foreground and background pixels respectively. Due to connectivity paradox, different connectivity should be used for the foreground and for the background. So either $\mathcal{G} = 8$ and $\mathcal{G}' = 4$, or inversely. Function $CC_{\mathcal{G}}$ returns the set of $\mathcal{G}$-connected components of its argument.

By the successive removal of simple pixels the image is thinned. Image obtained by this thinning performed till idempotence is a *skeleton* which - due to its homotopy preservation property - is also referred to as *homotopic marking*. Usually, thinning is performed in two stage iterative process. In the first phase the simple pixels are detected within the whole image (but not yet removed), while the proper removal of these pixels is performed in the second phase. These two phases are performed iteratively until thinned image stops to change.

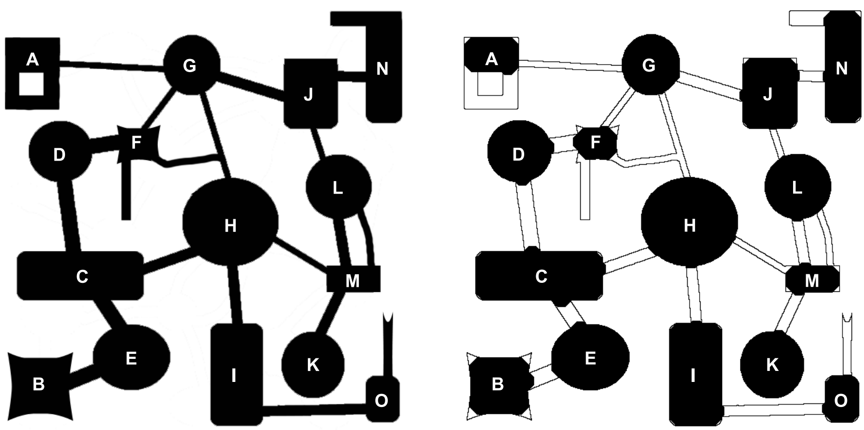
In order to have the possibility to control the thinning process, the notion of *anchor pixels* has been introduced [6]. These pixels are defined separately and - by definition - cannot be removed during the thinning process even if they are simple. The anchored skeletonization requires two input images - the image to be thinned and the image containing anchor pixels referred to as *anchor image*.

3 Graph Extraction Method

3.1 Detection of Objects and Connections - Graph Nodes and Edges

Images containing rasterized graph-like structures consist of two types of regions: regions referring to graph nodes (objects) and regions referring to graph edges (connections). In order to extract the graph structure from this image, at first, one should extract from input image pixels belonging to one of two classes: objects and connections.

At the beginning, objects are localized within the image. It is performed using the morphological opening operator. Usually, in majority of image processing applications, this operator is used to remove 'salt' noise. Side-effect which appears along with this removal is the modification of shape of image foreground. This feature, considered as disadvantage from filtering point of view, is a great advantage as far as the discussed task is taken into account. It allows detecting relatively thick object connected with relatively thin and elongated connections. The result of binary opening is often described as a region covered by structuring element in all possible positions totally included in the foreground of the input image. Consequently, thin fragments of of the foreground, thinner than the structuring element are not included in the result of opening. Among all such fragments of the image, also connections between objects are included. On the other hand, a majority of the surface of objects are present in the result of the opening. In case of image foreground structure resembling a graph, opening transforms in this way the single connected component of the foreground of input image into several connected components, each of which refer to single object - node of the graph. This is illustrated in Fig. 2, the result of an opening of the image on the left is shown on the right-hand side.



(a) (b)

Fig. 2. Input test image (a) and the result of of morphological opening (b)

The result of opening contains thus simplified shape of all objects. Formally, it can be described as:

$$I_O = (I \ominus B) \oplus B^T, \tag{2}$$

where I stands for the input image, I_O - image with simplified objects, B - structuring element and $B^T = \{p : -p \in B\}$ - for its transposition.

There exists direct relation between the structuring element used and the number of detected objects. The choice of structuring element is thus important since it determines which fragments of the foreground are considered as object. Consequently, it allows to find out which of them are graph nodes, and which are not. Generally, the bigger structuring element is, the lower number of object is extracted. This is due to the fact that opening with bigger structuring element(i.e. covering wider neighborhood) would cause the removal not only of connections but also of smaller potential objects.

In order to find connections between object, the residue of opening will be analyzed[1]. This residue contains all the fragments of foreground which was too thin to be able to include moving structuring element. Within these fragments, among others, also connections between objects are present. In order to find them, the anchored homotopic thinning of the input image is applied. Since the resulting skeleton - by definition - is homotopic to the original shape, the single connected component of a graph-like structure will be transformed into a single connected component of the skeleton. Moreover, if the result of opening will be considered as the set of anchor pixels, the skeleton will contain all the skeleton lines connecting them. Result of such skeletonizing is shown in Fig. 3(a). This result is used to compute simplified connections using the following equation:

$$I_C = (I_O - askel(I, I_O)) \oplus \mathcal{N}_8, \tag{3}$$

where I_C stands for simplified connections, $askel(I, I_O)$ is the anchored skeletonization of I with anchor image I_O.

An additional dilation in the Eq. 3 makes the skeleton lines thicker. Thanks to that the simplified connections image has a non-empty intersection with simplified objects one. This fact is used to detect the joints - places where connection is attached to the object. It is computed as simple intersection:

$$I_J = I_O \cap I_C. \tag{4}$$

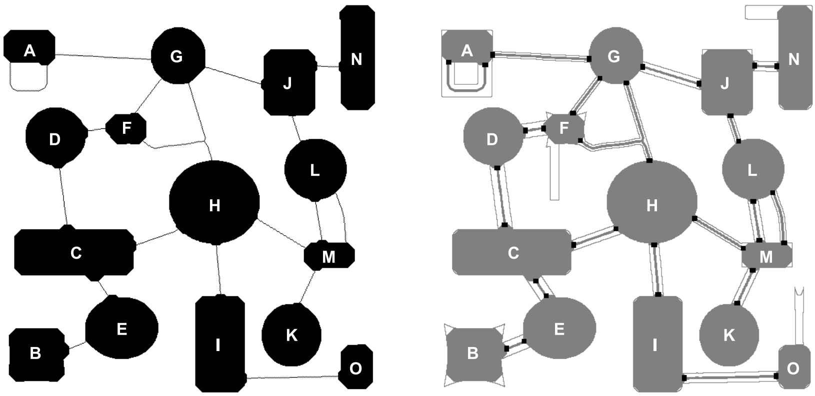
Detected joints on the test image are shown in Fig. 3(b). Joints are marked there in black, the simplified objects and connections as solid gray, the original foreground shape as gray outline.

Finally, each of three binary images: simplified objects, simplified connections and joints are labeled and three label image are created called: L_O, L_C and L_J, respectively. The number of connected components of I_O, I_C and I_J (or in other words, number of labels in L_O, L_C and L_J) will be denoted as n_O, n_C and n_J, respectively. Labeled images will be used to match objects, connections and joints and to produce to final adjacency matrix.

3.2 Adjacency Matrix Computation

Three labeled images obtained in the previous step are now processed in order to produce the adjacency matrix. The appropriate algorithm consists of two stages and is presented below.

[1] Residue of opening i.e. difference between original image and the result of opening is called white top-hat.



(a) (b)

Fig. 3. Result of anchored skeletonization of a test image (a) and result of joints detection (b)

Algorithm Adjacency matrix computation

$\mathcal{I}$	image definition domain
L_O,n_O	image with labeled objects (nodes) and the number of them
L_C,n_C	image with labeled connections (edges) and the number of them
L_J,n_J	image with labeled joints and the number of them
v_i	label of object (node) attached to i-th joint
e_i	label of connection (edge) attached to i-th joint
a	graph adjacency matrix of size $n_O \times n_O$

1. **for** $i \leftarrow 1, ..., n_J$ **do**
2.   **for** all $p \in \mathcal{I}$ **do**
3.   **if** $L_J(p) = i$ **then**
4. $v_i \leftarrow L_O(p)$ **;** $e_i \leftarrow L_C(p)$
5.   **break** the loop 2-5 and **goto** 1
6. **for** $i \leftarrow 1, ..., n_O$ **do**
7.   **for** $j \leftarrow 1, ..., n_O$ **do**
8. $a(i, j) \leftarrow 0$
9. **for** $i \leftarrow 1, ..., n_J - 1$ **do**
10.   **for** $j \leftarrow i + 1, ..., n_J$ **do**
11.   **if** $e_i = e_j$ **then**
12. $a(v_i, v_j) \leftarrow a(v_i, v_j) + 1$
13. $a(v_j, v_i) \leftarrow a(v_j, v_i) + 1$

In the first stage (lines 1-5) three images are analyzed and two temporary vectors v and e are produced. They contain information on the objects and connections attached to each joint: v_i is the label of object attached to i-th joint, while e_i stands for the label of connection attached to that joint.

In the second stage (lines 6-13) the graph adjacency matrix a is computed. First, matrix elements are set to 0 (lines 6-8). Next, all the joints are analyzed.

The analysis of each joint result in the increment of an appropriate element of matrix a. Since the graph - by definition - is undirected, the adjacency matrix is symmetric, so the increment is performed twice (lines 12 and 13). Since the increment is performed following appropriate joints, finally an element of the matrix contains the total number of connections (edges) between two objects (nodes).

Table 2. Graph adjacency matrix of graph shown in Fig. 2

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
A	2	0	0	0	0	0	1	0	0	0	0	0	0	0	0
B	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
C	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0
D	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
E	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	1	0	0	2	1	0	0	0	0	0	0	0
G	1	0	0	0	0	2	0	1	0	1	0	0	0	0	0
H	0	0	1	0	0	1	1	0	1	0	0	0	1	0	0
I	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
J	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0
K	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
L	0	0	0	0	0	0	0	0	0	1	0	0	2	0	0
M	0	0	0	0	0	0	0	1	0	0	1	2	0	0	0
N	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
O	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Example of graph adjacency matrix of an image shown in Fig. 2(a) is presented in Table 2.

4 Conclusions

In the paper a method for extraction from binary images, structures of the foreground, that resemble graph structures, was presented. Such foreground configurations consists of thick objects connected with thin and elongated connections. The objects and connections are detected using morphological image processing, in particular morphological opening (objects) and anchored homotopic skeletonization (connections). Depending on the structuring element used, various image regions can be classified as objects. Apart from object and connections, also their joints are detected. In the next step, based on the labeling of all these three images, the graph adjacency matrix is computed, which describes the structure of resulting graph.

The method can be applied in various areas of image processing where the graph-like structures are present on images under consideration. It may be used to describe these structures by providing with the graph adjacency matrix which can be used independently as final image description or as a feature for further recognition.

References

1. Iwanowski, M.: Binary Shape Characterization using Morphological Boundary Class Distribution Functions. In: Kurzynski, M., Puchala, E., Wozniak, M., Zolnierek, A. (eds.) Proc. of CORES 2007 Conference, October 22-25, 2007. Computer Recognition Systems 2 – Advances in Soft Computing, vol. 45. Springer, Heidelberg (2007)
2. Iwanowski, M.: Morphological Boundary Pixel Classification. In: Proc. of IEEE EUROCON Conference, Warsaw, pp. 146–150 (2007)
3. Soille, P.: Morphological image analysis. Springer, Heidelberg (1999) (2004)
4. Vincent, L.: Morphological grayscale reconstruction in image analysis: applications and efficient algorithms. IEEE Trans. on Image Processing 2(2) (April 1993)
5. Serra, J., Vincent, L.: An overview of morphological filtering. Circuit systems Signal Processing 11(1) (1992)
6. Vincent, L.: Efficient computation of various types of skeletons. In: Loew, M. (ed.) Medical Imaging V: Image Processing. SPIE, vol. 1445, pp. 297–311 (1991)
7. Vincent, L.: Exact Euclidean distance function by chain propagations. In: Proc. IEEE Computer Vision and Pattern Recognition, pp. 520–525 (1991)
8. Kong, T., Rosenfeld, A.: Digital topology: Introduction and survey. Computer Vision, Graphics, and Image Processing 48, 357–393 (1989)
9. Serra, J.: Image analysis and mathematical morphology, vol. 1. Academic Press, London (1983)
10. Rosenfeld, A.: Connectivity in digital pictures. Journal of the ACM 17(1), 146–160 (1970)