Directional Stroke Width Transform to Separate Text and Graphics in City Maps

Ali Ghafari-Beranghar [a,*], Ehsanollah Kabir [b], Kaveh Kangarloo [c]

[a] *Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran,*
[b] *Department of Electrical and Computer Engineering, Tarbiat Modarres University, Tehran, Iran,*
[c] *Department of Electrical Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran*

Received 4 January 2012; accepted 8 January 2013

Abstract

One of the complex documents in the real world is city maps. In these kinds of maps, text labels overlap by graphics with having a variety of fonts and styles in different orientations. Usually, text and graphic colour is not predefined due to various map publishers. In most city maps, text and graphic lines form a single connected component. Moreover, the common regions of text and graphic lines have similar features; hence, the separation of text and graphic lines is a challenging task in document analysis. Generally, these text labels could not be recognized efficiently by current commercial OCR systems in city map processing. In this paper, we propose an image decomposition approach based on stroke width feature to extract text labels from city maps. In our approach, we assign to each pixel of image a local stroke width based on minimum distance from borders in four directional borders. This mapping generates a suitable representation to distinguish text and non-text pixels. The experimental results on several varieties of city maps are promising.

Keywords: Text/Graphics separation, Directional Stroke Width, Graphics document processing, City map, Text segmentation

1. Introduction

One of the major problems in document image analysis is separating text from graphics. City maps are complex documents and separation of text from such document images is a challenging task due to variations of font, size, orientation and overlapping of texts and graphics lines. City maps are a common tool that enables people to find places and to better understand the features of a particular destination. Both government and business organizations must frequently convert existing paper maps of cities and larger regions into a machine-readable form that can be interfaced with current geographical information systems (GIS) or optical character recognition(OCR)[1]. Moreover, city maps are widely used by people for their travel planning and navigation needs, even for local and short trips[2]. City maps in raster format have rich information. Current OCR systems cannot recognize text labels in complex mixed text/graphics [1, 3, 4].

Commonly, text and graphics are separated for later analysis and recognition. Generally, text recognition is completely different from graphics recognition. Text labels in city maps can have various font types and sizes, touching or overlapping with graphics lines and in an arbitrary orientation. So, Text recognition from raster maps is a challenging task.

Most of the previous research focuses on geometry features of connected components including area, aspect ratio and density, so-called macro level features, to separate text component from graphics line[1, 5-8]. These features are fine to separate text and graphics that are not touching together. However, we focus on micro level features of connected components, i.e. pixels. Micro level features have more separating power than macro level ones. Macro level features for text and graphics are similar in the intersections of text and graphics. The aim of present work is to propose a local method to separate text from graphics in city maps. It computes a stroke width feature for each pixel of a connected component to differentiate between text and graphics. Experimental results show that, despite its simplicity, the proposed method works reasonably well in separating text from graphics. The proposed method provides a suitable framework to incorporate other features like intensity and colour to segment text from graphics efficiently.

Section 2 presents in details the related works. Section 3 explains the proposed method. Section 4 shows experimental results and section 5 analyzes error cases to

* Corresponding author. Email: a.ghafari.b@gmail.com

reveal some limitations of the proposed method. The conclusion is given in section 6.

2. Related Works

Text segmentation from graphics is an important step in document analysis. In the literature, researchers have proposed different methods to deal with this problem especially maps. Here, we briefly review these works with an emphasis on city maps.

To our knowledge, text/graphics separation methods can be divided to four main groups: connected component analysis, morphological operations, graph based analysis and distance based approach. In the literature, among these methods, connected component analysis has been widely used.

Using connected component method, Fletcher and Kasturi [5] proposed a popular algorithm to separate text from graphics. Hough transform was used to group the characters in a single word string. The main limitation of this method is the assumption that text and graphics are not touching together. Some researchers have tried to improve this algorithm. Tan and Ng [8] utilized a pyramid scheme instead of Hough transform to facilitate text segmentation. However, they also assumed that text is not touching to graphics. Other improvement was suggested by Tombreet al. [9] to use Fletcher and Kasturi's algorithm for complex graphics documents. They proposed a method to separate touching text from graphics line [10-12].

Morphological operations are also used for text/graphics segmentation. The structure element should be properly designed to separate text from graphics. This is not a simple task and depends on the input image[13, 14].

Graph-based method is commonly used as a representation tool to simplify text segmentation. Graph representation is extracted from the image and well-known graph-based algorithms are applied to separate text component from graphics [15-19]. The input image is converted to graph by using the thinning operation[17]. The thinning operation should meet the following requirements: (1) Resultant image skeletons from thinning operation must strictly have a single pixel width with eight neighbouring connection. (2) No disconnection should be created in thinning operation. (3) Inter-junctions of original text image components should be well preserved. (4) Corner points of original image components should be maintained [17]. Therefore, the main limitations of this approach are come from image-to-graph conversion requirements.

Some studies were reported in distance-based category. The notion of distance is fundamental in image analysis. Models for image representation generally involve minimum distances between pixels and borders of components. It is often the case that measurements are operated between a point and a set rather than between points. Such an operation defines an important tool in image analysis, the distance transformation [20]. In this category, some works have been reported that used distance in various forms.

Since stroke width is a good feature to describe text in graphical documents, some methods have been proposed to find the stroke width. For instance, [21] designed a content-based partition named as stroke width transform to extract text characters with stable stroke widths. In addition, the colour uniformity of text characters in natural scene image is taken into account for content-based partition[22]. An algorithm to find stroke width from distance map has been reported in [23].

All of these works have been reported in natural scene images. For example, stroke width feature has been used to extract text from scene images[21]. They used edges of binary image to find stroke width of each pixel. But our approach is not dependent to edge finding and so it is not sensitive to edge noises.

In this paper, we propose a new method based on stroke width. Stroke width has been adopted as a fine text descriptor in document analysis. In literature, stroke width has been mainly used at text component level. However, we use this technique in pixel level to classify text and graphics pixels.

3. Proposed Method

In document images like maps text and graphics lines can be seen as strokes drawn on a paper. Stroke width is a good feature for text and graphics description [24-26]. The proposed text/graphics Separation process is divided into 4 Steps: a) Binarization and component labelling, b) Directional stroke width transform (DSWT for short), c) image decomposition based on stroke width and d) Converting DSWT image to binary. The block diagram of text/graphics separation is given in Fig. 1. Part of input image and corresponding output, text and graphics image, is shown in Fig. 2. In the following sections, we explain each step.

a. Binarization and Component Labelling

The map images are digitized in gray level at 300 dpi. We use Otsu algorithm [27] to convert a gray level image map into binary, connected component labelling is performed on this binary image to detect individual components. Our method uses these components to assign stroke-width value to each pixel. We assume that components as foreground pixels are shown by "1" and background by "0".

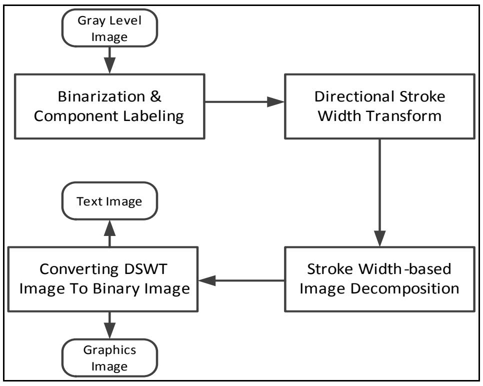

Fig. 1. Block diagram of text/graphics separation.

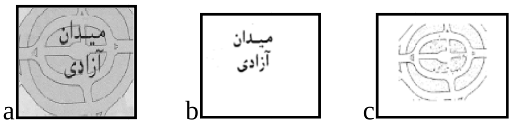

Fig.2. a) Original input image. b) Text image. c) Graphics image

b. Directional Stroke Width Transform

The Directional Stroke Width Transform, DSWT for short, is a local operator which estimates, per pixel, the width of the most likely stroke containing the pixel in certain orientations. The output of the DSWT is an image of size equal to the input image where each element contains the width of the stroke associated with corresponding pixel.

After finding connected components, we calculate the smallest of the four line segments passing through each point of component, p0, in four orientations, horizontal, vertical, diagonal 45 and diagonal 135 degrees (Fig. 3). This calculation has been described in detail in the following relation:

$$\forall p_0 \ni C, DSWT\,(p_0) = \min{}_{i=1}^{4}(\overline{P_{2i-1}P_{2i}}) \qquad (1)$$

Where, C is a connected component, p_0 is a point of C and $\overline{P_{2i-1}P_{2i}}$ is Euclidian distance between two borders P_{2i-1} and P_{2i} of the current component. So four distances isobtained.

We call the minimum of these distances as directional stroke width. We assign this value to the corresponding pixel in connected component. A connected component and its DSWT have been shown in Fig. 4(a), 4(b).

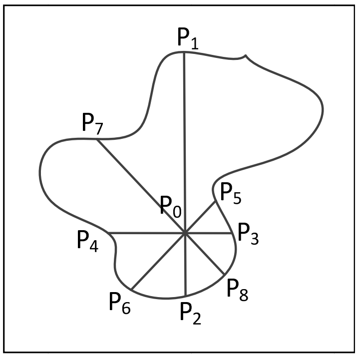

Fig. 3.A Connected Component C, P_1P_2 is vertical line segment, P_3P_4 is in horizontal and P_5P_6 is in diagonal 45 degrees and P_7P_8 is in diagonal 135 degrees. Euclidian Distance between P_{2i-1} and P_{2i} passing through P_0 is calculated for i = 1 to 4. DSWT in P_0 is the smallest length of $P_{2i-1}P_{2i}$ in four orientations. DSWT is the value of stroke width in corresponding point. In this figure, the length of P_5P_6 is the corresponding stroke width of P_0.

	1	1	1		
	1	1	1	1	
	1	1	1	1	
			1		
			1		
			1		

a

	1	3	3		
	3	3	3	2	
	1	3	3	2	
			1		
			1		
			1		

b

Fig. 4.a) A connected component. b) DSWT of image in (a).

3.3 Image Decomposition based on DSWT

After calculating DSWT, we have an image with the same size as original one. We observed that the graphics lines are thinner than the text associated with them in city maps. So we can decompose the DSWT image into text and graphics images based on a threshold. It can be calculated by finding least stroke width that has enough frequency due to other stroke width values. So, each pixel in DSWT whose value is greater than this threshold is classified into text and those lower are classified into graphics image. Therefore, we have two DSWT images that separate text and graphics pixel into two classes. For example, in Fig.

4(a) a connected component is shown. ‘1’ is for foreground pixels and background pixels are shown by empty white cells. DSWT of this connected component was calculated by relation (1) and is shown in Fig. 4(b). By histogram analysis, we select th = 2, so DSWT image is decomposed into text and graphics regions. In Fig. 5(a) text region of connected component and in Fig. 5(b) graphics region has been shown.

a

		3	3		
	3	3	3		
		3	3		

b

	1				
				2	
	1			2	
			1		
			1		
			1		

Fig. 5. a) Text region decomposed by Th= 2, b) Graphics region decomposed by Th = 2.

3.4 Convert DSWT Imageto BinaryImage

After decomposing text and graphics DSWT image, we want to obtain text and graphics binary image. To convert, we substitute each value of non-zero pixel to 1 or on pixel and zero pixels as background remain unchanged. At this point, text and graphics images are ready for further processing. In this paper, we have not done any post-processing.

4. Experimental Results

To our best knowledge, there is no standard published dataset to evaluate our method. So we gathered our own dataset under the following conditions. City maps were gathered from major publishers, Sahab Geographic and Drafting Institute and National Geographical Organization of Iran [13-14]. We took 50 city map images of size 1200 by 1000 by scanning parts of these maps at 300 dpi. We cut images into smaller ones of sizes 300 by 300 to 400 by 400. Various areas of images were selected to test our method. There were some city maps in bilingual format (Fig. 8a.).In city maps that we chose, as shown in figures 6a to 8a, text and graphics line fully overlap and street lines cross all characters of street names.

The scanned city maps were converted into binary images. We used Otsu’s method to convert data image to binary image. Here ‘0’ is used for background and ‘1’ is used for foreground objects. Most of city maps were in the Persian language and some of them are in English (Fig. 8a).

Text/graphics separation results for three sample city map are shown in figures 6-8. Here, the original image is shown (Fig. 6a-8a). Text and graphics images are shown without any post processing. In figures 6b-8b, text extracted from original image is shown.

In figures 6c to 8c, graphics lines extracted from original image are shown.

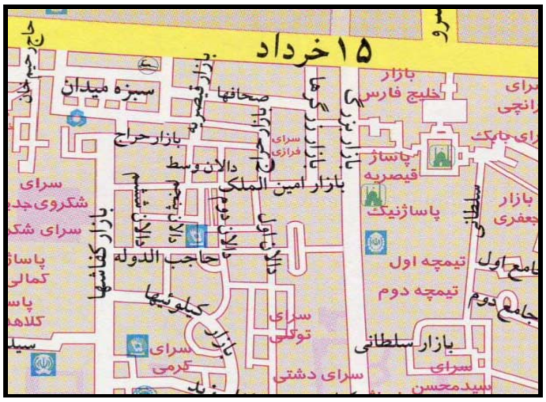

Fig. 6a: Part of a city map: Grand Bazaar of Tehran

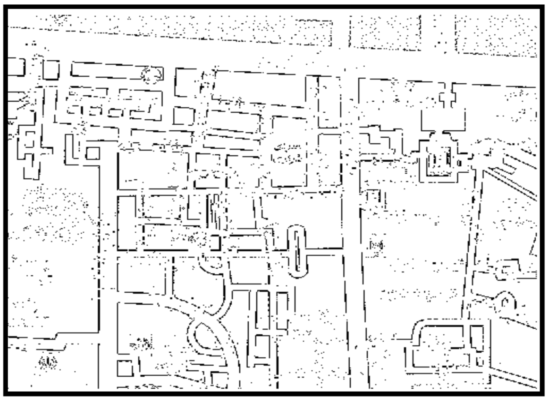

Fig. 6b: Graphics lines extracted from Fig. 6a.

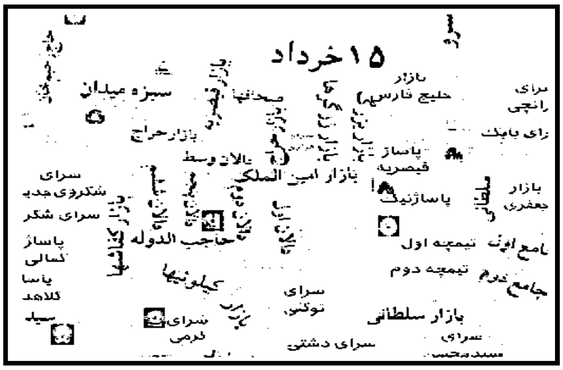

Fig. 6c: Text part extracted from Fig. 6a.

Fig. 7a: Part of a map: Highway Junction

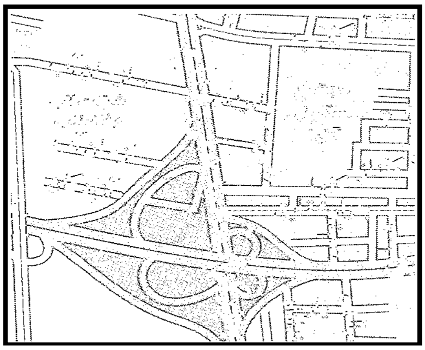

Fig. 7b: Graphics line separated from original image Fig. 7a.

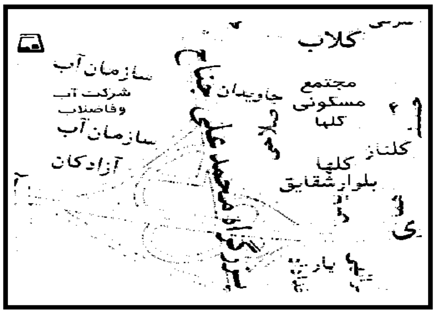

Fig. 7c: Text extracted from Fig. 7a.

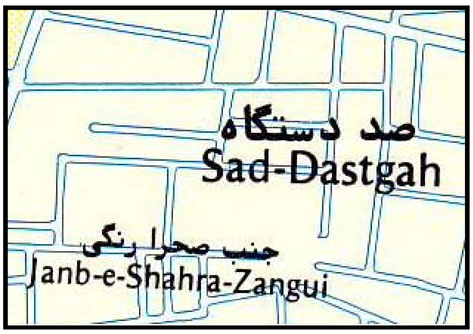

Fig. 8a: Part of a bilingual map.

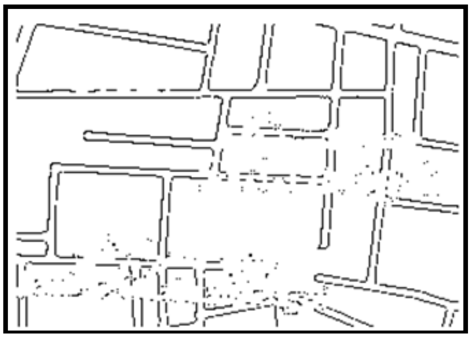

Fig. 8b: Graphics separated from Fig. 8a.

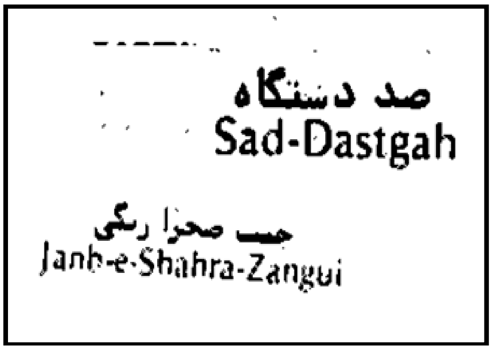

Fig. 8c: Text extracted from Fig. 8a.

5. Discussion

The main goal of this work was to design a method to separate text from graphics line in city maps. In city maps that we chose, as shown in figures6a - 8a, text and graphics line fully overlap and street lines cross all characters of street names. So connected component based methods could not be applied properly to tackle this problem. In fact, connected component based approach can separate text and graphics in situations that text and graphics components are disjoint and geometrical features of components can distinguish between text and graphics objects. But in city maps, text labels and graphics line are not separable by geometrical features of connected components.

In our method, a pixel-based feature instead of features at connected component level is used. We use stroke width feature locally in pixel level instead of connected component level. Our method is capable of assigning pixel of mixed text and graphics connected components into text and graphics classes.

We assume that text stroke width and average line width are locally distinguishable by stroke width feature. In other words, text stroke is thicker than graphics line width. In some cases where character stroke is thin so that its width is close to line width, our method has errors. In this case, these weak pixels of text are classified as graphics and text character is broken.

One important thing in our method is to select stroke width threshold in text/graphics separation. It depends on average line width ofimage you deal with. Commonly, in city maps, we observed that line width is thinner than text stroke width. In our experiments, we used 1 or 2 as a suitable value for separation of graphics line from text labels. Of course, text stroke width is not constant especially in pixel level that we have used. So choosing further a high value for stroke width causes more text stroke is classified as graphics (see figures 6b-8b).For example, some pixels of text have been incorrectly classified in graphics (see Fig. 7b). This error is due to a minimum text stroke width. We assume that the minimum stroke width of the text is greater than graphics. In some cases, this assumption is not true; therefore, our method has some error.

6. Conclusion

Text/graphics separation problem has been considered in fully overlapping situations. The text and graphics lines fully overlap in our experimental data set. We designed a method based on stroke width feature to separate text and graphics. Our method is language independent. Of course, the proposed method was tested for Persian and English languages.

Despite the simplicity of the proposed method, it is suitable for separating text from graphics in complex situations like city maps. In many applications, text or graphics with the predefined thickness should be extracted. In such applications, the proposed method seems to work well. Therefore, the proposed method has the ability to generalize.

References

[1] Li, L., et al., *Integrated text and line-art extraction from a topographic map.*International Journal on Document Analysis and Recognition, 2000.2(4): pp. 177-185.

[2] Wang, Z., N. Li, and B. Li, *Fast and Independent Access to Map Directions for People Who are Blind.*Interacting with Computers, 2012.24(2): pp. 91-106.

[3] Zhong, D.X., *Extraction of embedded and/or line-touching character-like objects.*Pattern Recognition, 2002.35(11): pp. 2453-2466.

[4] oy, P.P., et al., *A System to Retrieve Text/Symbols from Color Maps using Connected Component and Skeleton Analysis.*Relation 10, no. 1.96, 2008.

[5] Fletcher, L.A. and R. Kasturi, *A robust algorithm for text string separation from mixed text/graphics images.*Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1988.10(6): pp. 910-918.

[6] Kasturi, R. and J. Alemany, *Information Extraction from Images of Paper-Based Maps.*Software Engineering, IEEE Transactions on, 1988.14(5): pp. 671-675.

[7] Ebi, N., B. Lauterbach, and W. Anheier, *An image analysis system for automatic data acquisition from colored scanned maps.*Machine Vision and Applications, 1994.7(3): pp. 148-164.

[8] Tan, C. and P. Ng, *Text extraction using pyramid.*Pattern Recognition, 1998.31(1): pp. 63-72.

[9] Tombre, K., et al., *Text/graphics separation revisited*, in *Document Analysis System V*, E.P. Lopresti, J. Hu, and R. Kashi, Editors. 2002. pp. 200-211.

[10] Tan, C.L., et al. *Text/graphics separation using agent-based pyramid operations*. in*Proceedings of the Fifth International Conference on Document Analysis and Recognition, ICDAR '99.* . 1999. Bangalore.

[11] Cao, R. and C.L. Tan. *Separation of overlapping text from graphics*.in*Sixth International Conference on Document Analysis and Recognition*. 2001. Seattle, WA.

[12] Cao, R. and C.L. Tan, *Text/graphics separation in maps*, in *Graphics Recognition Algorithms and Applications*. 2002. pp. 167-177.

[13] Luo, H., G. Agam, and I. Dinstein, *Directional mathematical morphology approach for line thinning and extraction of character strings from maps and line drawings*, in *Proceedings of the Third International Conference on Document Analysis and Recognition*. 1995, IEEE Computer Society: Montreal, Que. pp. 257-260.

[14] Hasan, Y. and L. Karam, *Morphological text extraction from images.*Image Processing, IEEE Transactions on, 2000.9(11): pp. 1978-1983.

[15] Biswas, S. and A.K. Das. *Text extraction from scanned land map images*. in*International Conference on Informatics, Electronics & Vision (ICIEV)*. 2012.

[16] Hoang, T.V. and S. Tabbone. Text extraction from graphical document images using sparse representation. in Proceedings of the 9th IAPR International Workshop on Document Analysis Systems. 2010. New York, NY, USA: ACM.

[17] Cheng, Z. and Y. Liu, *A Graph-based Method to Remove Interferential Curve From Text Image.*Machine Vision and Applications, 2006.17(4): pp. 219-228.

[18] Cheng, Z., M. Chen, and Y. Liu, *A robust algorithm for image principal curve detection.*Pattern Recognition Letters, 2004.25(11): pp. 1303-1313.

[19] Raveaux, R., J.C. Burie, and J.M. Ogier. *A colour text/graphics separation based on a graph representation*. in*19th International Conference on Pattern Recognition*. 2008. Tampa, FL.

[20] Marchand-Maillet, S. and Y.M. Sharaiha, *Binary digital image processing: a discrete approach*. 2000: Academic Press.

[21] Epshtein, B., E. Ofek, and Y. Wexler. *Detecting text in natural scenes with stroke width transform*.in*IEEE Conference on Computer Vision*

and Pattern Recognition (CVPR). 2010. San Francisco, CA: IEEE.

[22] Chucai, Y. and T. YingLi, *Text String Detection From Natural Scenes by Structure-Based Partition and Grouping.*Image Processing, IEEE Transactions on, 2011.20(9): pp. 2594-2605.

[23] Chen, H., et al. *Robust text detection in natural images with edge-enhanced maximally stable extremal regions*. in*18th IEEE International Conference on Image Processing (ICIP)*. 2011. Brussels.

[24] Shi, J., N. Ray, and H. Zhang, *Shape Based Local Thresholding for Binarization of Document Images*.Pattern Recognition Letters, 2012.33(1): pp. 24-32.

[25] Nguyen, T.T.H., M. Coustaty, and J.M. Ogier, *Bags of Strokes Based Approach for Classification and Indexing of Drop Caps*, in *International Conference on Document Analysis and Recognition*. 2011: Beijing. pp. 349-353.

[26] Bataineh, B., S.N.H.S. Abdullah, and K. Omar, *An adaptive local binarization method for document images based on a novel thresholding method and dynamic windows.*Pattern Recognition Letters, 2011.32(14): pp. 1805-1813.

[27] Otsu, N., *A threshold selection method from gray-level histograms.*Automatica, 1975.11(285-296): pp. 23-27