Heuristics-Based Detection to Improve Text/Graphics Segmentation in Complex Engineering Drawings

Carlos Francisco Moreno-García(✉), Eyad Elyan, and Chrisina Jayne

The Robert Gordon University, Garthdee Road, Aberdeen, UK
c.moreno-garcia@rgu.ac.uk

Abstract. The demand for digitisation of complex engineering drawings becomes increasingly important for the industry given the pressure to improve the efficiency and time effectiveness of operational processes. There have been numerous attempts to solve this problem, either by proposing a general form of document interpretation or by establishing an application dependant framework. Moreover, text/graphics segmentation has been presented as a particular form of addressing document digitisation problem, with the main aim of splitting text and graphics into different layers. Given the challenging characteristics of complex engineering drawings, this paper presents a novel sequential heuristics-based methodology which is aimed at localising and detecting the most representative symbols of the drawing. This implementation enables the subsequent application of a text/graphics segmentation method in a more effective form. The experimental framework is composed of two parts: first we show the performance of the symbol detection system and then we present an evaluation of three different state of the art text/graphic segmentation techniques to find text on the remaining image.

Keywords: Complex engineering drawing · Digitisation · Text/graphics segmentation · Connected component analysis

1 Introduction

We define a complex engineering drawing (CED) as any type of schematic diagram which aims at representing the flow or constitution of a circuit, a process, a plant or a device. Unlike the classical definition of an engineering drawing (ED) which includes standard logical gate circuits, mechanical or architectural drawings, through this new definition we intend to characterise a specific subset of EDs with a complexity that demands a more advanced series of methods for their automated digitisation. Some examples of CEDs are process and instrumentation diagrams (P&IDs), chemical process diagrams, complex circuit diagrams, telephone manholes and facility drawings. An example of a portion of a P&ID is shown in Fig. 1.

G. Boracchi et al. (Eds.): EANN 2017, CCIS 744, pp. 87–98, 2017.
DOI: 10.1007/978-3-319-65172-9_8

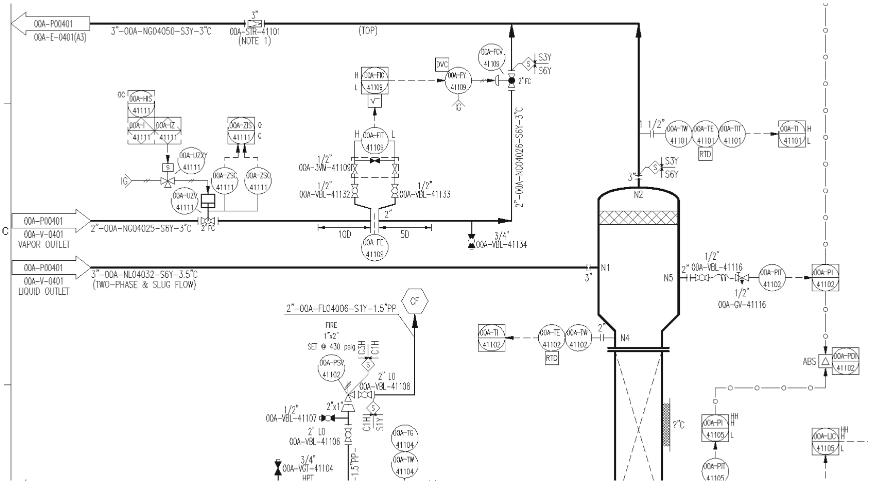

Fig. 1. Example of a portion of a process and instrumentation diagram

While ED digitisation has been largely reviewed and addressed in literature [1,2,7,8,15,20–22,27,34] the digitisation and contextualisation of CEDs still conveys several problems such as:

- *Size*: A single page of a CED contains about 100 different types of representations and around 150 symbols. Moreover, several pages (100 to 1000) may be required to represent a single process or structure.
- *Symbols*: Besides the conventional problem of variability in size, direction and position of symbols, CEDs present different symbol standards for different industries, and even when comparing two CEDs designed within a same company, standards may vary due to time. This leads to the constant employment of new symbols to describe incompatible elements. Hence, creating a symbol repository for training purposes [15] is sometimes not a viable solution.
- *Connections*: CEDs contain a dense and entangled structure of different types of connectors which represent physical and logical relations between symbols. Various type of connectors are usually depicted using different sizes and thickness, and thus methods based on thinning [27] become limited.
- *Text*: It is common that CEDs contain a large amount of codes and annotations (printed or handwritten). Moreover, connectors may also have corresponding text which contains important information such as the width of a connector. In general, CEDs are filled with a considerable amount of text which must be identified as well, since it is key for symbol recognition and drawing contextualisation.

Given the importance of text detection for CED digitisation and contextualisation, a particular kind of methodology called text/graphics segmentation can be considered for this task. For the past 35 years, it is possible to identify a vast amount of literature related to text/graphics segmentation methods for document images [9,12,13,17,24,30,31]. These methods may have a general purpose or be directed to a certain application domain, such as maps [5,29], book pages [6,32] and EDs [4,23]. While the characteristics of CEDs difficult the straightforward application of these methods, if a robust preprocessing and segmentation of symbols and connectors is applied to the CED in advance, then text/graphics segmentation becomes a viable option to locate the text remaining on the image.

The rest of the paper is structured as follows. In Sect. 2, we present the related work in text/graphics segmentation. Section 3 presents our proposed methodology to address the problem at hand. Section 4 contains the description of our experimental setting and the discussion of the results. Finally, Sect. 5 is reserved for conclusions and future work.

2 Related Work

In [1], Ablameyko and Uchida performed a review on ED digitisation methods, focusing on methods to detect symbols, connectors and text. Authors denote that most methods separate text from graphics either before or after ED vectorisation. Moreover, they noticed that text is commonly identified by using heuristics which help the system select either single characters or complete strings through certain constraints such as size, directional characteristics or complexity. Once text is isolated, the system either groups characters as strings within a certain space or erodes non-character shapes to keep text only. Most recently in [33], Wei et al. published a study on methodologies used for text detection in outdoor scene images. Authors found that text detection in most domains require two steps: character segmentation and string grouping. The first task is usually addressed through region-based methods, connected component (CC) analysis [28] or hybrid methods, while the second one is solved through a rule set approach, a clustering method or by learning algorithms. So far in literature, text/graphic segmentation methods mostly rely on CC analysis for character segmentation and rule-based string grouping.

In 1988, Fletcher and Kasturi [13] presented an algorithm to find text in printed drawings regardless of position, orientation or size of the text. The method consists on first applying CC analysis to the drawing in order to locate each character and graphic, discarding the ones longer than a size threshold and a height-to-width ratio threshold. To group characters into strings, authors introduce a methodology for linear analysis based on applying the Hough transform [18] to the centroids of the text CCs, which is a widely used method that has been applied to find lines [11,25], arbitrary shapes [3] and in more recent work, to locate partial images within their full counterparts [26]. This system has become a largely replicated solution due to its versatility and simplicity, however one of its greatest disadvantages is the incapability of the system to correctly identify individual characters and text overlapping lines or even other characters.

In 1998, Lu [23] presented a text/graphic separation method for characters in EDs. This method aims at erasing non-text and graphics from the ED to leave text only. Authors proposes a series of rule-based steps consisting on (1) erasing large line components, (2) erasing non-text shapes by analysing stroke density instead of size constraints and (3) grouping character into strings through a brush and opening operation to form new CCs, followed by a second parameter check on this newly formed CCs which restores miss-detected characters into their respective strings. The method deals better with the problem of text overlapping lines since most characters are left on the image and can be recovered on the last step. However, this method is very prone to identify false positives (such as small components or curved lines) and depends on text strings to be apart from each other so that the string grouping is executed correctly.

In 2002, Tombre et al. [31] presented an upgrade on [13] for document images rich in graphics. Authors increase the number of constraints on the first step of the original method so that the best enclosing rectangle of a shape identified as text is considered before analysing the CC. In addition, since the density and the elongation of the CCs are calculated and analysed for the text/graphic distinction, authors create a third layer where small elongated elements (i.e. “1”, “—”, “l”, “-” or dot-dash connectors) are stored. At the second stage of the algorithm, authors propose alternative strategies to compute the string grouping in the Hough transform domain, which according to the characteristics of the document image, could lead to better or worse results. Finally authors add an extension of string step where shapes on the small elongated element space are restored into the text space into their respective strings according to an analysis of proximity. Other interesting papers that present improvements on CC analysis based text/graphics segmentation are He and Abe [17], where clustering is used to improve each step, Tan and Ng [30] where a pyramid version of the image is used to group strings, or Chowdhury et al. [6] that proposes a multi decision tree for a more specific segmentation.

Regarding work for text detection in other areas, the method for outdoor scenes proposed by Wei et al. [33] is based on an exhaustive segmentation approach, which means that multiple image binarisations are generated from a single image using the minimum and maximum gray pixel value as threshold range. Then, candidate character regions are determined for each binarisation based on CC analysis, and non-character regions are filtered out through a two-step strategy composed of a rule set and a Support Vector Machine (SVM) classifier working on a set of features i.e. area ratio, stroke-width variation, intensity, Euler number [16] and Hu moments [19]. After combining all true character regions through clustering, authors implement an edge cut approach for string grouping. This consists of first establishing a fully connected graph of all characters, and then calculating the true edges based on a second SVM classifier using a second set of features i.e. size, colour, intensity and stroke width. This method clearly results in a more complex and robust approach to the problem at hand, however it is difficult to implement on printed drawings. Nonetheless, methods of this nature lead us to realise that there are interesting alternatives to the classical text/graphics segmentation methods in literature.

3 Methodology

We propose a sequential heuristics-based methodology which is aimed at localising and removing the most representative symbols of the drawing, with the aim of preparing the image for the use of a text-graphics segmentation method which can detect text characters across the remaining image. In summary, the complete CED digitisation framework consists on the following steps:

1. Preprocessing.
2. Image resizing.
3. Detection of representative shapes.
   (a) Linear components.
   (b) Connectivity symbols.
   (c) Remaining geometrical symbols.
4. Text/Graphics Segmentation.

3.1 Heuristics-Based Symbol Detection

After applying preprocessing methods such as thresholding and noise removal, it has been noticed that several CEDs are surrounded by a blank frame which increase the size of the file and hence the time for digitisation. To discard this outer frame automatically, we apply a Canny edge detector on the image. Then, the resulting image is dilated using a cross structural element recursively, intending to connect all the schematics contained. Finally, a CC analysis is run, and only the portion of the original drawing located on the bounding box of the dilated image is considered as the input of the system.

While most text/graphics segmentation methods suggest to discard indistinctively all lines larger than a certain threshold (usually dependent on the average character height) either by analysing CCs of a large width or height [13] or by scanning the image for large sequences of pixels across different image inclinations [23], in CEDs large lines represent different aspects of the drawing based on their length and thickness. For instance, in P&IDs there are thick and long lines that represent pipelines or the outline of a vessel, thick and short lines that represent smaller symbols such as emergency shut down valves, and thin and long lines that represent the margin line, connectors, symbols. Therefore, a more thoughtful detection methodology has been implemented so that each long line is correctly localised within its context and thus the classification complexity can be reduced. To do so, the first aim is to detach symbols and text from connectors and large elements. Given h and w representing the height and the width of the input image respectively, the image is dilated two separate times using a rectangular structuring elements of size $(1 \times h/m)$ and $(w/m \times 1)$ respectively. Variable m must be set to a high value (i.e. one third of the size of the longer edge of the image) to allow the horizontal and vertical lines to be maintained to the most on each image after the dilation. Then, both images are combined to create a new image containing only large lines. The pixels of the input image which are not included in this image are considered either part of other symbols or text. Afterwards, a blur operation is applied to the resulting image so that the thicker lines can be distinguished from the thin ones. Thick line segments are analysed as follows; if one or more thick line segments conform a loop, then this is a representation of a thick line symbol/vessel, otherwise this line represents a connector. Searching for loop elements in an image can be addressed through several means, such as finding the Euler number [16], an enclosing chain code [14] or by contour detection. Regarding thin lines, these are classified according to their localisation. If the line is long and close to the image border, then it is a margin line; otherwise it is a connector line.

The next step is to locate symbols which are characteristic of the drawing and that have properties which allow their detection in more efficient manners. Such is the case of continuity labels, which are text-boxes in the end of thick connector lines which indicate the connection of the represented piping to another drawing. Since these labels are located on either side of the schematic, they can be located by scanning the new image either applying template matching or CC analysis. Given that these labels contain text, it is recommended to segment continuity labels along with the contained text for this to be used on later stages. For instance, a learning methodology could be applied to analyse this text in advance and deduce the average text size so that the subsequent text/graphics segmentation step can be automated and thus more effective.

Finally, geometrical symbols such as circles and polygons can be located. Circles may be found within dot-dash connectors or representing symbols, and can be segmented through the application of the Hough circles method [3] taking into consideration factors such as size and localisation to avoid false positives within text. On the other hand, polygons can be detected through contour detection and approximation, by means of methods such as the Douglas-Peucker algorithm [10]). If these instrumentation symbols contain text, this creates a second opportunity to read text or learn its properties in advance.

3.2 Text/Graphics Segmentation

Once the image without the aforementioned symbols and connectors has been generated, a text/graphics segmentation method can be applied. The main aim at this stage is to distinguish characters and group text strings considering the following limitations:

- *Symbols and connectors left*: Long or dashed lines representing connectors or measuring indicators, as well as symbols such as grid areas, irregular or disconnected polygons e.g. arrowheads, diamonds, trapezoids, or loop free symbols (a capacitor or a resistance) may still be present in the diagram. Examples are shown in Fig. 2a.
- *String size irregularity*: The characters to be grouped into strings present irregular shapes and sizes. While in some cases the string is vertical or horizontal, some others it is split into rows. Furthermore, in some cases a symbol splits the string either top-bottom or left-right in Fig. 2b, different string shapes and sizes are shown.

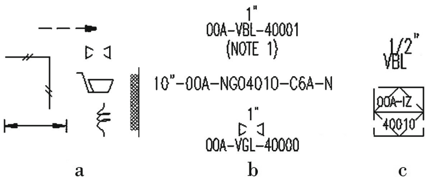

Fig. 2. Examples of limitations found in a P&ID once symbols found through the heuristics-based method have been removed. (a) Symbols and connectors left, (b) string size irregularity and (c) character overlapping

- *Punctuation signs*: It is of particular interest to avoid discarding punctuation sings (i.e. inches symbols, periods and commas) without wrongfully identifying them as noisy components
- *Character Overlapping*: Some of the text characters may overlap symbols and connectors, or even other characters, as shown in Fig. 2c.

To that aim, the most widely used text/graphics segmentation methods used on EDs are based on CC analysis. As described in Sect. 2, once CC analysis is done, characters and graphics are split into separate layers according to different variables according to the application and the system design, such as area, height-to-width ratio, stroke density, pixel density, elongation, number of loops, etc. Furthermore, some other methods create a third layer to store elongated elements (i.e. letters "l" and "i" or symbols "-" and "/").

4 Experiments and Results

After describing the drawings that compose the dataset used for experimentation, we briefly report the results of applying the heuristics-based methodology on the drawings to detect and segment as many representative symbols as possible. Then, we compare the character detection effectiveness of three state of the art text/graphics segmentation methods on either the original images of the dataset or on the image after the application of the heuristics-based symbol detection. This way, we aim to verify that detecting and segmenting the most representative symbols in advance leads to an improvement in the character detection.

4.1 Dataset Used

We have compiled from an industrial partner a collection of P&ID drawings with a large and dense quantity of symbols, connectors and text (an example of these drawings is shown in Fig. 1). Images have been scanned at a 300 dpi resolution resulting on average in 3508 × 2479 pixels size. In total, the drawings on the collection contain an average of 2.9 thick line symbols/vessels, 41 circular instrumentation symbols, 32.7 circles within dot-dash connectors, 6.6

continuity labels, 34.6 polygons (triangles, squares and hexagons) plus tenths of other irregular and unclassified symbols. Furthermore, drawings contain over one hundred text strings each, which range from 1 to 24 characters of length and may be grouped in different shapes and extensions according to the process or instrumentation described.

4.2 Heuristics-Based Method

After preprocessing, the image is reduced on average 85.28% from its original size. Afterwards, the image is inspected for line components. This system successfully distinguishes pipeline connectors (blue), margin lines (purple) and vessels (cyan) as shown in Fig. 3.

Successively, the method is capable of finding all continuity labels for all drawings easily, (red box in Fig. 3a). Furthermore, with the proper tuning of the radius we are capable of locating all large circles representing two types of instrumentations (red and yellow circles in Fig. 3b). Moreover, an average of 96.52% of small circles within dot-dash connectors are detected (light green circles in Fig. 3c) by using the circle detection method plus deducing the location of the missed small circles by analysing the dot-dash connector itself. To that aim, the image is scanned and all small linear segments adjacent to small circles (dark green) are identified as connectors. Using the small circles and line segments, the path of the dot-dash connector can be constituted and the missing circles are located. Also, circles within symbols such as valves can be found by using a similar approach. Finally, the polygon detection algorithm based on contour detection is capable of locating an average of 83.32% of the squares, diamonds and triangles on the datasets. Notice that this is the least accurate

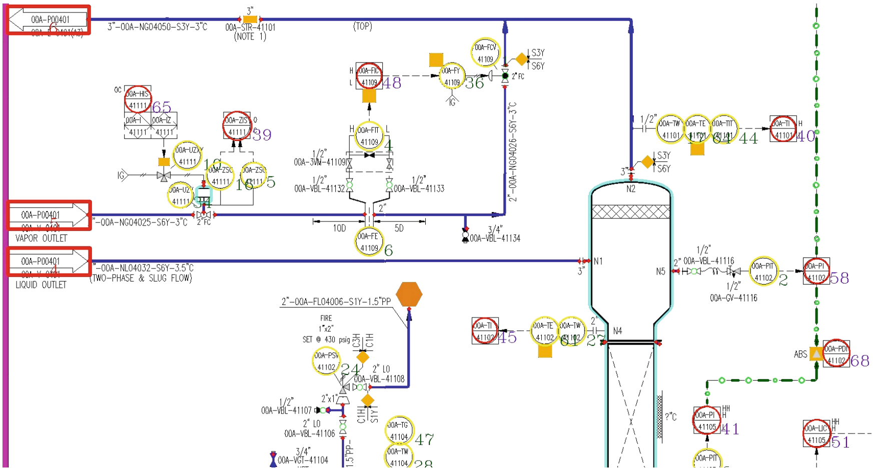

Fig. 3. Result of applying the sequential heuristics-based methodology on Fig. 1 (Color figure online)

of the detectors since this step depends on polygon approximation methods and thus, many of these symbols are not correctly approximated once their contours are detected.

4.3 Comparison of Character Detection in Text/Graphics Segmentation Methods

We have tested the character detection feature of three text/graphics segmentation methods in literature: Fletcher and Kasturi [13], Lu [23] and Tombre et al. [31]. These methods have been selected since they are the base of most existing methods and because they present the two step approach described in Sect. 2 (character detection and string grouping) and thus this enables a fair comparison.

In order to test whether the inclusion of the heuristics-based symbol detection method leads to an improvement and to compare the accuracy of the text/graphics segmentation methods, we have applied these methods both without and with the previous application of the heuristic-based symbol detection on our P&ID dataset. We present in Table 1 a comparison of precision, recall and runtime for the six possible scenarios.

Table 1. Comparison of accuracy (precision and recall) and average runtime between the character detection methods without or with the previous application of the heuristics-based symbol detection.

	No heuristics-based detection			With heuristics-based detection		
Character detection	Precision	Recall	Runtime (s)	Precision	Recall	Runtime (s)
Fletcher and Kasturi [13]	0.37	1	0.93	0.91	0.98	12.14
Lu [23]	0.39	1	14.88	0.93	0.98	18.35
Tombre et al. [31]	0.47	1	234.61	0.98	0.84	185.72

With respect to the accuracy, notice that the three methods present the highest possible recall given that they are capable of including all existing text; however since a large amount of false positives are included, precision is reduced. In contrast, the character detection methods after the application of the heuristics-based detection present high precision while mantaining a good recall, considering that in steps identify text. Notice that the precision using [31] is lower than in the other cases given that we are not considering the small elongated components that have yet to be classified as text or graphics during the string grouping phase.

Regarding the runtime to compute the full process, notice that the first two cases delay more when performing heuristics-based segmentation plus character detection than when applying character detection only. However, it can be appreciated that if [31] is used, it is less time consuming to apply both processes rather than applying character detection on the original image. This occurs because at the character detection stage there are less CCs to analyse after the symbol detection has taken place. Therefore, we infer that when applying more robust text/graphics segmentation methods using more complex filtering, applying a previous symbol detection could not only lead to an improvement in accuracy, but also in the runtime of the system overall. Tests where performed using a PC with Intel 3.4 GHz CPU and Windows 10 operating system.

5 Conclusions

In this paper, we present a symbol detection method aimed at improving the application of text/graphics segmentation on CEDs. This method uses an heuristic-based approach to detect and segment the most representative symbols of the drawing, using as example the case of a P&ID. We have tested our system on a collection of drawings with a large and dense quantity of symbols, connectors and text, and we have noticed that a high amount of symbols can be recognised if the algorithm is properly set and the characteristics of the drawing are understood in advance. Moreover, we have performed a comparison between different state of the art methods that perform character detection on engineering drawings. We apply three character detection methods in two cases: on the original image or on the image after the heuristics-based symbol detection has been applied. We have seen that the character detection after the symbol detection outperforms the application on the original image, since less false positives are detected and less strings have to be processed. Furthermore, the average runtime of applying each scenario has been calculated, noticing that for the most robust text/graphics segmentation method, an improvement in runtime can be achieved if the symbol detection is applied beforehand.

There is a clear room for further work in this area, given the large need for digitisation systems for CEDs. Firstly, we aim at completing the text/graphics segmentation process and test different grouping strings methodologies. Also, we aim at considering more advanced heuristics which allow to overcome usual problems in CEDs such as character overlapping. Finally, we intend to test our proposed methodology in more datasets containing a wider range of symbols and characteristics.

Acknowledgement. We would like to thank Dr. Brian Bain from DNV-GL Aberdeen for his feedback and collaboration in the project. This work is supported by a Scottish national project granted by the Data Lab Innovation Centre.

References

1. Ablameyko, S.V., Uchida, S.: Recognition of engineering drawing entities: review of approaches. Int. J. Image Graph. **07**(04), 709–733 (2007)
2. Arias, J.F., Lai, C.P., Chandran, S., Kasturi, R., Chhabra, A.: Interpretation of telephone system manhole drawings. Pattern Recognit. Lett. **16**(4), 365–368 (1995)
3. Ballard, D.H.: Generalizing the Hough transform to detect arbitrary shapes. Pattern Recognit. **13**(2), 111–122 (1981)

4. Bunke, H.: Automatic interpretation of lines and text in circuit diagrams. In: Kittler, J., Fu, K.S., Pau, L.F. (eds.) Pattern Recognition Theory and Applications, vol. 81, pp. 297–310. Springer, Dordrecht (1982)
5. Cao, R., Tan, C.L.: Text/graphics separation in maps. In: Blostein, D., Kwon, Y.-B. (eds.) GREC 2001. LNCS, vol. 2390, pp. 167–177. Springer, Heidelberg (2002). doi:10.1007/3-540-45868-9_14
6. Chowdhury, S.P., Mandal, S., Das, A.K., Chanda, B.: Segmentation of text and graphics from document images. In: Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, vol, 2 (Sect. 4), pp. 619–623 (2007)
7. Cordella, L.P., Vento, M.: Symbol recognition in documents: a collection of techniques? Int. J. Doc. Anal. Recognit. **3**(2), 73–88 (2000)
8. De, P., Mandal, S., Bhowmick, P.: Identification of annotations for circuit symbols in electrical diagrams of document images. In: 2014 Fifth International Conference on Signal and Image Processing, pp. 297–302 (2014)
9. Dori, D., Wenyin, L.: Vector-based segmentation of text connected to graphics in engineering drawings. In: Perner, P., Wang, P., Rosenfeld, A. (eds.) SSPR 1996. LNCS, vol. 1121, pp. 322–331. Springer, Heidelberg (1996). doi:10.1007/3-540-61577-6_33
10. Douglas, D.H., Peucker, T.K.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartogr. Int. J. Geogr. Inf. Geovisualization **10**(2), 112–122 (1973)
11. Duda, R.O., Hart, P.E.: Use of the Hough transformation to detect lines and curves in pictures. Commun. ACM **15**, 11–15 (1971)
12. Fan, K.C., Liu, C.H., Wang, Y.K.: Segmentation and classification of mixed text/graphics/image documents. Pattern Recognit. Lett. **15**(12), 1201–1209 (1994)
13. Fletcher, L.A., Kasturi, R.: Robust algorithm for text string separation from mixed text/graphics images. IEEE Trans. Pattern Anal. Mach. Intell. **10**(6), 910–918 (1988)
14. Freeman, H.: On the encoding of arbitrary geometric configurations. IRE Trans. Electron. Comput. **EC–10**, 260–268 (1960)
15. Gellaboina, M.K., Venkoparao, V.G.: Graphic symbol recognition using auto associative neural network model. In: Proceedings of the 7th International Conference on Advances in Pattern Recognition, ICAPR 2009, pp. 297–301 (2009)
16. Gray, S.B.: Local properties of binary images in two dimensions. IEEE Trans. Comput. **20**(5), 551–561 (1971)
17. He, S., Abe, N.: A clustering-based approach to the separation of text strings from mixed text/graphics documents. Proc. - Int. Conf. Pattern Recognit. **3**, 706–710 (1996)
18. Hough, P.V.C.: Method and means for recognizing complex patterns. US Patent 3,069,654, 18 December 1962
19. Hu, M.K.: Visual pattern recognition by moment invariants. IRE Trans. Inf. Theory **8**, 179–187 (1962)
20. Kasturi, R., Bow, S.T., El-Masri, W., Shah, J., Gattiker, J.R.: A system for interpretation of line drawings. IEEE Trans. Pattern Anal. Mach. Intell. **12**(10), 978–992 (1990)
21. Kim, S.H., Suh, J.W., Kim, J.H.: Recognition of logic diagrams by identifying loops and rectilinear polylines. In Proceedings of the Second International Conference on Document Analysis and Recognition - ICDAR 1993, pp. 349–352 (1993)
22. Lladós, J., Valveny, E., Sánchez, G., Martí, E.: Symbol recognition: current advances and perspectives. In: Blostein, D., Kwon, Y.-B. (eds.) GREC 2001. LNCS, vol. 2390, pp. 104–128. Springer, Heidelberg (2002). doi:10.1007/3-540-45868-9_9

23. Lu, Z.: Detection of text regions from digital engineering drawings. IEEE Trans. Pattern Anal. Mach. Intell. **20**(4), 431–439 (1998)
24. Luo, H., Agam, G., Dinstein, I.: Directional mathematical morphology approach for line thinning and extraction of character strings from maps and line drawings. In: Proceedings of 3rd International Conference on Document Analysis and Recognition, vol. 1, pp. 257–260, 1 August 1995
25. Matas, J., Galambos, C., Kittler, J.: Robust detection of lines using the progressive probabilistic hough transform. Comput. Vis. Image Underst. **78**(1), 119–137 (2000)
26. Moreno-García, C.F., Cortés, X., Serratosa, F.: Partial to full image registration based on candidate positions and multiple correspondences. In: Bayro-Corrochano, E., Hancock, E. (eds.) CIARP 2014. LNCS, vol. 8827, pp. 745–753. Springer, Cham (2014). doi:10.1007/978-3-319-12568-8_90
27. Okazaki, A., Kondo, T., Mori, K., Tsunekawa, S., Kawamoto, E.: Automatic circuit diagram reader with loop-structure-based symbol recognition. IEEE Trans. Pattern Anal. Mach. Intell. **10**(3), 331–341 (1988)
28. Pratt, W.K.: Digital Image Processing, 4th edn. Wiley, Los Altos (2013)
29. Roy, P.P., Vazquez, E., Lladós, J., Baldrich, R., Pal, U.: A system to segment text and symbols from color maps. In: Liu, W., Lladós, J., Ogier, J.-M. (eds.) GREC 2007. LNCS, vol. 5046, pp. 245–256. Springer, Heidelberg (2008). doi:10.1007/978-3-540-88188-9_23
30. Tan, C., Ng, P.O.: Text extraction using pyramid. Pattern Recognit. **31**(1), 63–72 (1998)
31. Tombre, K., Tabbone, S., Pélissier, L., Lamiroy, B., Dosch, P.: Text/graphics separation revisited. In: Lopresti, D., Hu, J., Kashi, R. (eds.) DAS 2002. LNCS, vol. 2423, pp. 200–211. Springer, Heidelberg (2002). doi:10.1007/3-540-45869-7_24
32. Wahl, F.M., Wong, K.Y., Casey, R.G.: Block segmentation and text extraction in mixed text/image documents. Comput. Graph. Image Process. **20**(4), 375–390 (1982)
33. Wei, Y., Zhang, Z., Shen, W., Zeng, D., Fang, M., Zhou, S.: Text detection in scene images based on exhaustive segmentation. Signal Process. Image Commun. **50**, 1–8 (2017)
34. Yu, Y., Samal, A., Seth, S.C.: A system for recognizing a large class of engineering drawings. IEEE Trans. Pattern Anal. Mach. Intell. **19**(8), 868–890 (1997)