# Text block segmentation in comic speech bubbles

Christophe Rigaud, Nhu-Van Nguyen, Jean-Christophe Burie

# Text block segmentation in comic speech bubbles

Christophe Rigaud[0000−1111−2222−3333]
Nhu-Van Nguyen[0000−0003−0291−0078]
Jean-Christophe Burie[0000−0001−7323−2855]

Laboratoire L3i, SAIL joint laboratory
Université de La Rochelle
17042 La Rochelle CEDEX 1, France
{christophe.rigaud, nhu-van.nguyen, jean-christophe.burie}@univ-lr.fr

**Abstract.** Comics and manga text recognition are attracting an increasing research and industrial interest. Also, the state of the art text detection and OCR performances is starting to be mature enough to provide automatic text recognition for a variety of comics and manga writing styles. However, comics text layout sometimes prevents usual text line detection to be applied successfully, even within speech bubbles. In this paper, we propose a domain specific text block detection method able to detect single and multiple text block regions inside speech bubbles, in order to enhance OCR transcription and further post-processing. This approach presents very satisfactory results on all tested bubble styles from Latin and non-Latin scripts.

**Keywords:** text block detection · layout analysis · comics analysis.

## 1 Introduction

Comics around the world are following some commonly accepted rules and a lot of particularities [14]. In this paper we are tackling one of them related to speech bubble content layout. Speech bubbles (or speech balloons) are specific "easy to read" regions were most of the text of the story is encapsulated. Most of the time there is a single block of text within each speech bubble but sometimes there are several blocks. For instance, when one character has multiple balloons within a panel, often only the balloon nearest to the speaker's head has a tail, and the others are connected to it in sequence, sometimes using narrow bands (Fig. 1). In the first case, a classical OCR (Optical Recognition System) system is able to recognize characters, words and text lines if appropriately trained. However, in the case of sequential text block contained by several inter-connected speech bubbles, text lines can get mixed up between text block and the corresponding output transcription not following the natural reading order anymore (Fig. 2).

Multiple text blocks within speech bubble might be absent, rare or frequent depending on the album. For instance, in the eBDtheque dataset [8], even if there are not annotated they can be considered as rare because we counted only 21 over 1081 total speech balloon occurrences over its hundred images. In

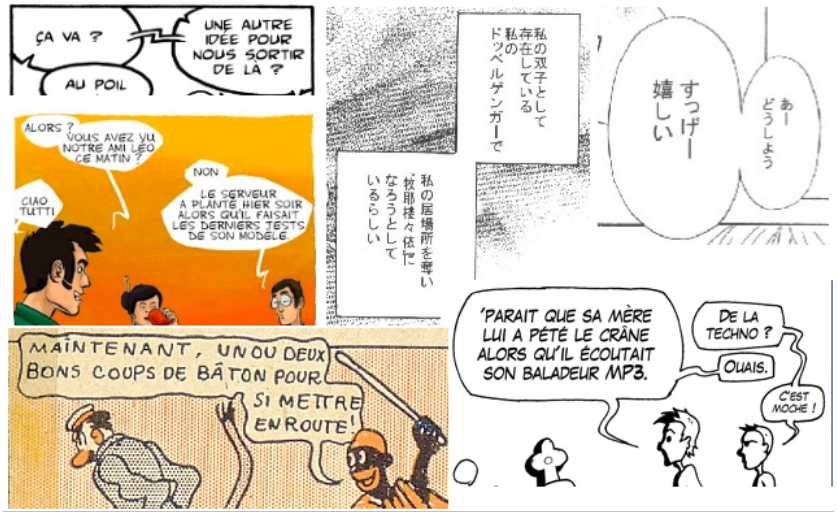


**Fig. 1.** Examples of connected bubbles. Image credits from eBDtheque [8]: *Cyb - Bubblegom p.36, Inoue Kyoumen p.6-13, Lubbin - Les bulles du labo p.1, Zig et Puce - Millionaire p.15 and Lamisseb - Et Pis Taf p.13.*

Manga109 dataset [2], there seems to be more frequent but as there are not annotated neither. Unfortunately, we couldn't count them all manually in this dataset because they are too many annotated text boxes (147,918). On another hand, in some volumes of Belgian comics like XIII[1] or Thorgal[2], private albums shared by a partner from our SAIL joint laboratory, we visually counted 1 to 3 inter-connected bubble per page.

When such bubble configuration appears on almost every page, it becomes essential to correctly detect the blocks of text, without mixing up text lines of nearby blocks. Especially, when one is interested in the automatic text recognition (OCR), translation or speech synthesis (with coherent reading order). This process is part of a global understanding process consisting of segmenting regions like panel, bubble, text, comic character and "connecting" them all together. For instance, after a complete understanding, it would be possible to retrieve that "Hello Bob" is written in bubble $A$ which should be read before the connected bubble $B$ containing text "How are you?", both said by the comic character $C$ to the comic character $D$ and all these four elements are contained by panel $E$ in page $F$ of album $G$.

In order to contribute on this specific issue, we review the related literature in Section 2, propose a first approach in Section 3, present our results in Section 4 and conclude this work in Section 5.

[1] https://en.wikipedia.org/wiki/XIII_(comics)
[2] https://en.wikipedia.org/wiki/Thorgal


**Fig. 2.** Example of mixed up text line detection (left) and the corrected detection using the proposed text block segmentation (right). Image credits: *XIII (Mystery) - vol. 11, Dargaud, p. 19.*

## 2 Related works

Text block segmentation has been largely studied in the literature and seems to be considered as solved for many types of documents. The most used technique for segmenting words, text lines and text blocks from letters (connected components) is Run Length Smoothing Algorithm (RLSA) [21]. This algorithm has been originally designed for Manhattan document layout analysis and rely on horizontal/vertical thresholds. Indeed, too high thresholds could erroneously merge different blocks in the document, while too low ones could return an excessively fragmented layout. Later approaches have attempted to set these thresholds automatically for Manhattan and non-Manhattan layout (RLSO) [7,20]. This approach has been applied to newspaper, scientific article, trademarks and administrative documents sometimes handwritten [22,4].

Another algorithm called Docstrum [16] consist in a bottom-up approach based on the nearest neighborhood clustering of connected components extracted from the document image.

Almost a decade later, Delaunay triangulation and its dual graph the Voronoi diagram have started to be applied to document segmentation [10,11]. The general problem they are trying to solve using Voronoi method is the following: given a set of element centers in the plane (center of letters, words or text lines), they associate with each element a region consisting of all points in the plane closer to that element than any other element. Then, a threshold is applied to segment blocks far enough away. It relates to inter-character spacing and inter-word/line spacing thresholds. These can be determined statistically from the document page and their effects is well described in [1].

A comparison of the above-mentioned methods is compiled in the following paper comparing performance of six-page segmentation algorithms [19].

Since the Convolutional Neural Network (CNN) era, researchers have used the RLSA method to build text blocks from scientific papers and classify them using bi-dimensional CNN [3]. Also, a recent work applied an on-the-shelf fully convolutional neural network (EAST) [24] to detect text in comic books but without considering text block regions [17].

Most of the works presented in the above literature are highly application-related which makes them hardly applicable to comic books because they often have a specific and variable layout. Moreover, the reviewed methods have been tested on Latin script and their effectiveness on non-Latin (complex) scripts like Japanese, Arabic, Indian and Korean is not obvious [12]. Most of them require application-related thresholds that can difficultly be learned from comic book image or speech bubble themselves because they contain very few words compared to other documents. We remarked that comics share some characteristics with free style documents such as posters, business card, envelope etc. [23]. For instance, isolated text and complex background.

Another related study proposed a method for computing speech bubble reading order but did not consider the underlying text block separation/ordering within speech bubbles [9].

From our knowledge, there isn't any method from the literature presenting results about text blocks segmentation within speech bubbles. Therefore, in the next section we present a first adaptive and parameter-free method able to separate text blocks in speech bubbles with horizontal or vertical text.

## 3 Proposed approach

The proposed approach focuses on segmenting text blocks from speech bubbles more precisely than previous methods in order to be able to also segment multiple text blocks within the same bubble (when existing). This precise segmentation can enhance, for example, the straightforward text recognition step by avoiding text line mixing and also accurate reading order computing.

Most of the text being located in the speech bubble, we propose to rely on this domain-specific feature because it has been demonstrated that domain knowledge can boost text segmentation performances [6].

The specific multiple text block layout of these speech balloons makes previous method not straightforward applicable. Moreover, it currently prevents from using any training-based method because of the lack of training data. From our knowledge, publicly available datasets provide multiple text line (eBDtheque) or single text block (Manga109) location annotations within speech bubbles but none of them provide multiple text block annotation (in connected balloons).

Even though there are not as frequent as single text block speech bubbles, they may be a key issue for some specific albums as previously mentioned in the introduction. Note that the proposed approach is designed to be independent from text language and script, and assumes that speech bubble has been previously segmented. One interested by segmenting speech bubbles can use any methods from the literature such as [5,13,15,18].

Regarding the literature, the Voronoi approaches seems to be the most appropriate for separating text blocks but it requires several parameters such as inter-letter space. On another hand, speech bubble segmentation methods from the literature do not guarantee to detect multi text block bubble as separated bubbles so text line lines may be confused in the subsequent text recognition

step. To tackle this challenge, we preferred to propose a new adaptive approach based on each speech bubble content (robust to script and letter size changes).

The proposed method is best when applied within previously segmented speech bubbles. However, it can also be applied on the full image with some extra processing depending on the comics layout. The method consists in three steps:

1. Content detection
2. Bounding box enlarging
3. Text block detection

The proposed process is illustrated in Fig. 4 from top to bottom, we detail it in the next three paragraphs.

**Content detection** The first step consists in detecting all connected components contained inside bubble region using for instance connected components labelling (also called blob extraction). Then we compute their bounding boxes to be used in next step (see red boxes and corresponding masks in the Fig. 4).

**Box enlarging** The objective it to merge these bounding boxes in order to form a single region surrounding all letters from the same text block and not overlapping other text blocks. To do so, we magnify the width and the height of previously computed bounding boxes pixel by pixel and centered on the original box. Then, we analyse the evolution on the number of contours in the corresponding mask and stop the enlargement just at the beginning of the period when the number of components remains stable the longest before getting down to zero. In Fig. 4, its start with 38 contours and the longest period before zero is the one with 2 contours where the process automatically stops (see bar graph in the Fig. 3).

The growing region is limited to balloon contour, this means the set of enlarged boxes is cropped afterward if getting out. At that end, this should form a homogeneous region surrounding contained text blocks without touching any other.

**Text block detection** We detect the external contours within the corresponding mask and label them as text block regions. Once the text blocks are extracted, they can be sent to an OCR system in place of the speech balloon. By sending properly segmented text blocks instead of speech bubbles to the OCR system, we avoid text line confusions.

After these three steps, the text blocks can be analysed independently by, for instance, an OCR system and text line detection and reading order should no longer get mixed up. Note that in this very simple example, other trivial methods could have been effective, such as horizontal/vertical histogram projection, but this last can not be generalised to more complex examples that will be presented in the next section.

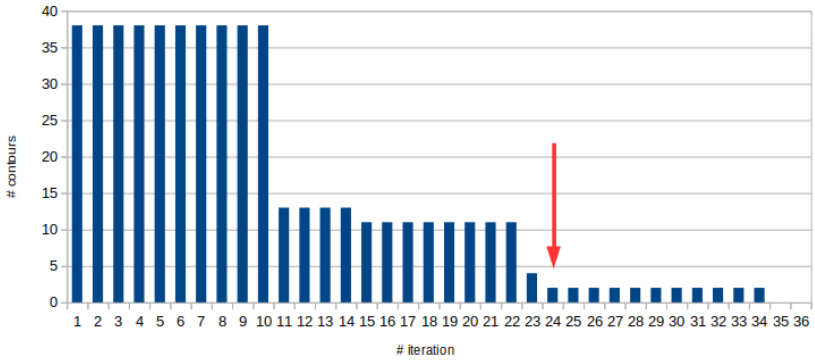


**Fig. 3.** Number of detected contours according to the number of iterations of bounding box enlarging and stop point (red arrow).

## 4 Results

We evaluated the proposed method for text block segmentation on several albums of Latin and Japanese scripts. Despite the ground truth, we performed only a qualitative evaluation on comics images from three datasets: eBDtheque (Fig.5), Manga109 (Fig.6) and other private albums (Fig. 7). In these three illustrations, the speech bubble segmentation shown with green line is used as prerequisite of the proposed method (text block segmentation). The detected text line bounding boxes are represented with a red line and are results from Tesseract OCR text line segmentation module (not part of the proposed method). There are shown only to visually measure the impact of one of the most frequent post-processing. Only the text block segmentation shown with a blue line are results from the proposed method.

### 4.1 eBDtheque dataset

Qualitative results on some images from the eBDtheque dataset are presented in Fig.5. They consist in different speech bubbles from a French webcomic on the left, a Japanese manga in the center and two printed French comics on the right-hand side. Unfortunately, we did not find any connected balloon of American comics style in image from this dataset. We consider the results on the webcomic as perfect because the three computed text block polygons are surrounding each of the three text blocks contained in the connected bubble, without touching any of the text letter (which is a common source of error for OCR system). The center manga bubble is also considered as a perfect result for the same reasons, even if the script is totally different from Latin. The third column bubbles shows some errors. Two of them are due to a missed character from the bubble segmentation algorithm (green line) which had difficulties to process these border-free bubbles in this low definition image. Another cause of error due to very close character

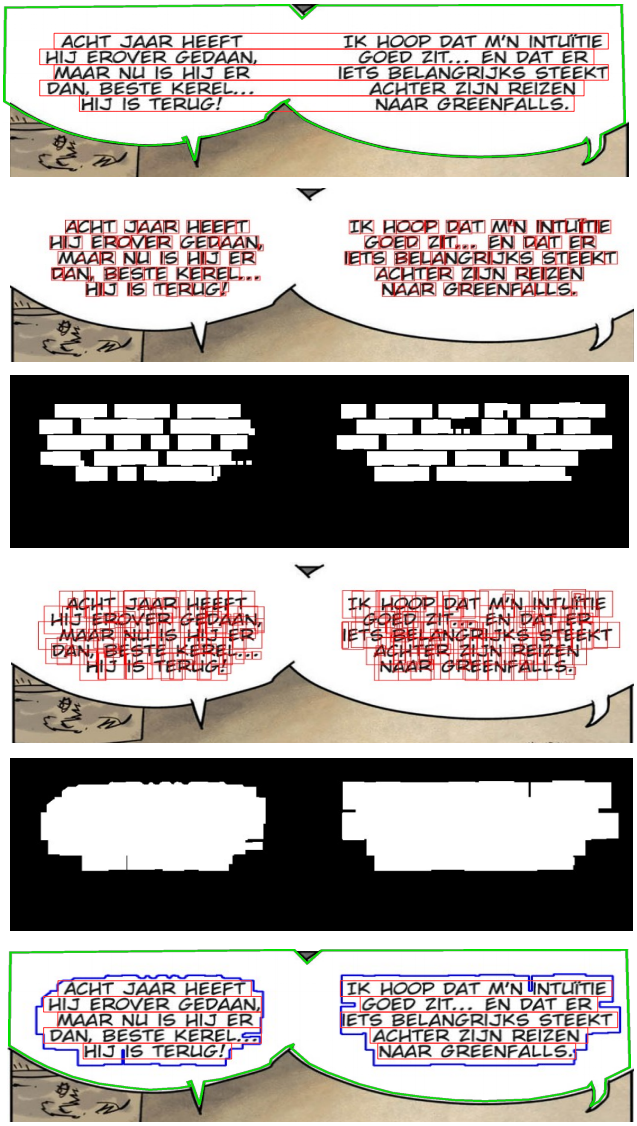


**Fig. 4.** Text block segmentation steps: from top to bottom, original erroneous text line detection (red) inside their properly segmented balloon (green); connected component (CC) bounding box detection (red) and corresponding mask; final enlarged CC and mask; text block detection (blue) and text line detection within text block. Image credits: *XIII (Mystery) - vol. 11, Dargaud, p. 19.*

from different lines of text is visible in the bottom-right image where there are no extra spaces between the second and the third text line.

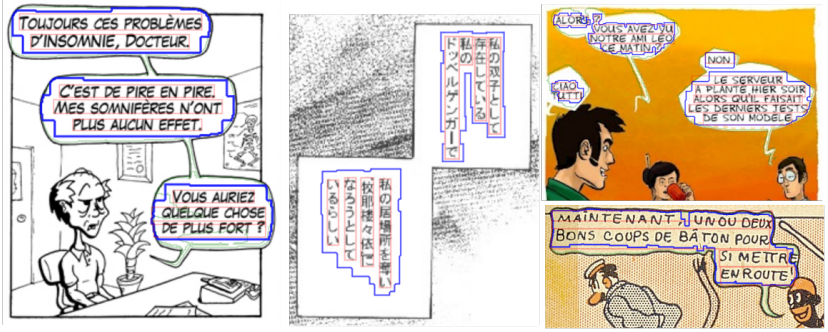


**Fig. 5.** Correct (left and center columns) and wrong (right column) results on eBDtheque dataset in French and Japanese (digitized images). Image credits from eBDtheque: *Lamisseb - Et Pis Taf p.5, Inoue Kyoumen p.6, Lubbin - Les bulles du labo p.1 and Zig et Puce - Millionaire p.15.*

### 4.2 Manga109 dataset

Qualitative results on some images from the Manga109 dataset are presented in Figure 6. They are selected from two random titles from Manga109 dataset: Seishinki Vulnus 2 and Platinum Jungle 1. Even if the Japanese script is more complex than Latin script, the proposed method still performs well for text block detection. Vertical text lines are well grouped into text blocks will juxtaposed Furigana character as well (which sometime appear no detected by the text line detection algorithm used has illustrative post-processing here). In the bottom-right corner, we added an interesting single block speech bubble where spaced dots have been segmented as separated text blocks by mistake. This is due to their successive small size that did not successfully merge with other surrounding components during the enlargement step.

### 4.3 Private dataset

We evaluated our method on other images from private albums in Dutch and French (non-shareable due to copyright). These images are from recent digital born Franco-Belgium comics and therefore has the advantage to be very clean and avoids any error due to image degradation. On the left half of Fig. 7, all text blocks have been well detected, even the "..." at the end of some sentences. However, on the two top bubbles, there are an exaggerated extension on the left side of the text block. This is due to the touching letters of the word "MAAR" which produces an extension equivalent to half of the word size instead of half of the letter size, as on the remaining of the text block outline.

The two connected balloons in the right column have not been well separated which makes mixed line detection (same as without text block detection). In each block, there are some letters or punctuation symbols from the two connected bubbles that are very close, relatively to their size, and the proposed method did not succeed in separating them.

**Fig. 6.** Vertical text line detection before (left) and after (right) text block detection in Manga109 dataset in Japanese (digitized images). Image credits from Manga109: *Seishinki Vulnus 2 (Yuzuru Shimazaki) and Platinum Jungle 1 (Masami Shinohara).*

### 4.4 Synthesis

We qualitatively evaluated the proposed method over few images from diverse datasets in order to demonstrate its robustness. All the reported errors have different importance depending on the bubble layout. For instance, in vertical Japanese script, if a text block is at the right or on the left of another text block it will probably not have any impact on the post-processing step because the text line detection will not get mixed anyway. However, if it is located above or below another text block within the same bubble, the text lines from the two blocks have more chances to be merged by text line detection algorithms writing (see Fig.8). Note that in all the experiments we performed, we observed that the enlargement step mainly stops between 2 and 4 times the original bounding box width and height (inter-dependent) e.g. $width * 2$ and $height * 2$.

## 5 Conclusion

In this paper we proposed a simple, fast, parameter-free and efficient domain-specific method for segmenting single and multiple text blocks from single and

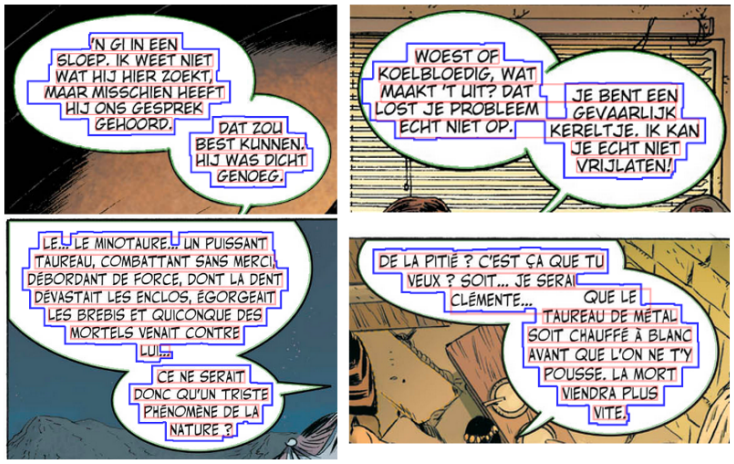


**Fig. 7.** Correct (left columns) and wrong (right column) results on private albums in Dutch and French (digital-born images). Image credits: *XIII (Mystery) - vol. 12, Dargaud, p. 30, 44 and Le Feu de Thesee (Survivre) - Les Humanoïdes Associés, p. 5, 18.*

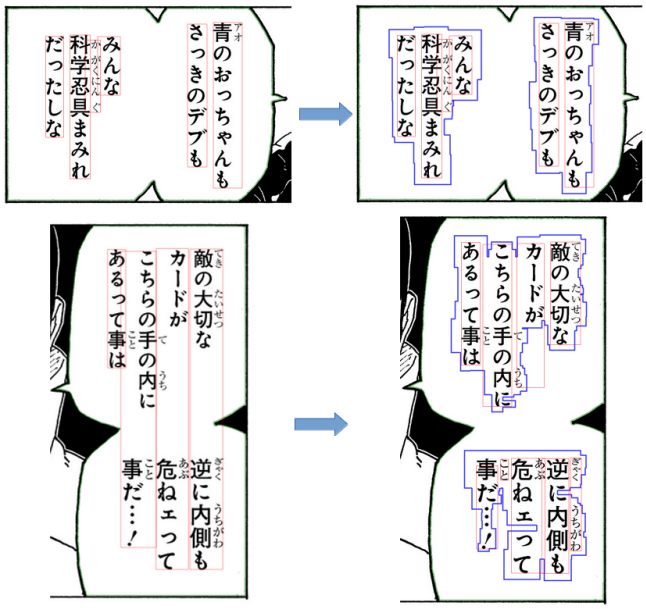


**Fig. 8.** Examples of vertical text line detection before (left) and after (right) text block detection for vertically and horizontally connected bubbles in Japanese manga. Image credits: *Boruto - vol. 7, Kana, p. 82, 103.*

connected speech bubbles. Even if connected speech bubbles are not so frequent, this method requiring little processing can easily complement other techniques such as speech balloon segmentation, comics OCR system and speech synthesis, to boost their results when connected speech bubble appear. We tested it only within pre-processed speech bubbles but we believe that it is not mandatory and will do more experiments in this direction. The proposed method turned out to be more suitable for detached writing (non-cursive) because it is based on connected component analysis and word size is much more variable than character size which alters the precision of the boundary of the computed blocks. Also, it works very well in the great majority of tested bubbles but sometimes fails when text blocks are very close (spaced by less than a letter/symbol size).

In the future, we plan to carry out experiments on other scripts like Arabic, Indian, Korean, etc. Also, we would like to remove the width and height inter-dependence to better fit text blocks, especially for non-Latin scripts. The annotation of existing comic book image datasets with precise text block positions and corresponding text line transcriptions would allow quantitative results and comparisons.

## 6 Acknowledgements

This work is supported by the Research National Agency (ANR) in the framework of the 2017 LabCom program (ANR 17-LCV2-0006-01), the CPER NUMERIC program funded by the Region Nouvelle Aquitaine, CDA, Charente Maritime French Department, La Rochelle conurbation authority (CDA) and the European Union through the FEDER funding.

## References

1. Agrawal, M., Doermann, D.: Context-aware and content-based dynamic voronoi page segmentation. In: Proceedings of the 9th IAPR International Workshop on Document Analysis Systems. pp. 73–80 (2010)
2. Aizawa, K., Fujimoto, A., Otsubo, A., Ogawa, T., Matsui, Y., Tsubota, K., Ikuta, H.: Building a manga dataset "manga109" with annotations for multimedia applications. IEEE MultiMedia **27**(2), 8–18 (2020). https://doi.org/10.1109/mmul.2020.2987895
3. Augusto Borges Oliveira, D., Palhares Viana, M.: Fast cnn-based document layout analysis. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 1173–1180 (2017)
4. Barlas, P., Adam, S., Chatelain, C., Paquet, T.: A typed and handwritten text block segmentation system for heterogeneous and complex documents. In: 2014 11th IAPR International Workshop on Document Analysis Systems. pp. 46–50. IEEE (2014)
5. Dubray, D., Laubrock, J.: Deep cnn-based speech balloon detection and segmentation for comic books. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 1237–1243. IEEE (2019)

6. Fan, K.C., Liu, C.H., Wang, Y.K.: Segmentation and classification of mixed text/graphics/image documents. Pattern Recognition Letters **15**(12), 1201 – 1209 (1994). https://doi.org/https://doi.org/10.1016/0167-8655(94)90110-4, http://www.sciencedirect.com/science/article/pii/0167865594901104
7. Ferilli, S., Leuzzi, F., Rotella, F., Esposito, F.: A run length smoothing-based algorithm for non-manhattan document segmentation. Convegno del Gruppo Italiano Ricercatori in Pattern Recognition **2012** (2012)
8. Guérin, C., Rigaud, C., Mercier, A., Ammar-Boudjelal, F., Bertet, K., Bouju, A., Burie, J.C., Louis, G., Ogier, J.M., Revel, A.: eBDtheque: A representative database of comics. In: 2013 12th International Conference on Document Analysis and Recognition. pp. 1145–1149 (Aug 2013)
9. Guérin, C., Rigaud, C., Bertet, K., Revel, A.: An ontology-based framework for the automated analysis and interpretation of comic books' images. Inf. Sci. **378**, 109–130 (2017). https://doi.org/10.1016/j.ins.2016.10.032, https://doi.org/10.1016/j.ins.2016.10.032
10. Kise, K., Sato, A., Iwata, M.: Segmentation of page images using the area voronoi diagram. Computer Vision and Image Understanding **70**(3), 370–382 (1998)
11. Koo, H.I., Cho, N.I.: State estimation in a document image and its application in text block identification and text line extraction. In: European Conference on Computer Vision. pp. 421–434. Springer (2010)
12. Kumar, K.S., Kumar, S., Jawahar, C.: On segmentation of documents in complex scripts. In: Ninth International Conference on Document Analysis and Recognition (ICDAR 2007). vol. 2, pp. 1243–1247. IEEE (2007)
13. Liu, X., Li, C., Zhu, H., Wong, T.T., Xu, X.: Text-aware balloon extraction from manga. The Visual Computer **32**(4), 501–511 (2016)
14. McCloud, S.: Understanding comics: The invisible art. Northampton, Mass (1993)
15. Nguyen, N.V., Rigaud, C., Burie, J.C.: Multi-task model for comic book image analysis. In: International Conference on Multimedia Modeling. pp. 637–649. Springer (2019)
16. O'Gorman, L.: The document spectrum for page layout analysis. IEEE Trans. Pattern Anal. Mach. Intell. **15**(11), 1162–1173 (Nov 1993). https://doi.org/10.1109/34.244677, https://doi.org/10.1109/34.244677
17. Rayar, F., Uchida, S.: Comic text detection using neural network approach. In: International Conference on Multimedia Modeling. pp. 672–683. Springer (2019)
18. Rigaud, C., Burie, J.C., Ogier, J.M.: Text-independent speech balloon segmentation for comics and manga. In: International Workshop on Graphics Recognition. pp. 133–147. Springer (2015)
19. Shafait, F., Keysers, D., Breuel, T.: Performance evaluation and benchmarking of six-page segmentation algorithms. IEEE transactions on pattern analysis and machine intelligence **30**, 941–54 (07 2008). https://doi.org/10.1109/TPAMI.2007.70837
20. Sun, H.M.: Page segmentation for manhattan and non-manhattan layout documents via selective crla. In: Eighth International Conference on Document Analysis and Recognition (ICDAR'05). pp. 116–120. IEEE (2005)
21. Wahl, F.M., Wong, K.Y., Casey, R.G.: Block segmentation and text extraction in mixed text/image documents. Computer Graphics and Image Processing **20**(4), 375 – 390 (1982). https://doi.org/https://doi.org/10.1016/0146-664X(82)90059-4, http://www.sciencedirect.com/science/article/pii/0146664X82900594
22. Wang, D., Srihari, S.N.: Classification of newspaper image blocks using texture analysis. Computer Vision, Graphics, and Image Processing **47**(3), 327 –

352 (1989). https://doi.org/https://doi.org/10.1016/0734-189X(89)90116-3, http://www.sciencedirect.com/science/article/pii/0734189X89901163
23. Xiaolu, S., Changsong, L., Xiaoqing, D., Yanming, Z.: Text line extraction in free style document. Proceedings of SPIE. L **72470** (2009)
24. Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., Liang, J.: EAST: an efficient and accurate scene text detector. CoRR **abs/1704.03155** (2017), http://arxiv.org/abs/1704.03155