Detection and Localization of Embedded Subtitles in a Video Stream

Grishkin Valery(✉) and Sene Jean

Saint-Petersburg State University, St.Petersburg 199034, Russia
v.grishkin@spbu.ru

Abstract. Videos with superimposed external subtitles constitute the major part of modern video content. However, there are quite a lot of diverse videos with embedded subtitles as well. In this regard, the problem arises of extracting and converting embedded subtitles into modern formats of external subtitles. Important steps in solving this problem are detection, localization, and binding of these subtitles to frames of the video stream. This paper proposes a method for detection and localization of embedded subtitles in video stream. The method is based on the search for static regions in the frames of the video stream and subsequent analysis of the connected areas inside them. Based on the results of this analysis, we determine whether a region belongs to the area of subtitles and localize text strings in the detected subtitles. The proposed method does not require large computational costs and can work in real time.

Keywords: Embedded subtitles · Subtitles localization · Image segmentation

1 Introduction

Currently, multimedia content makes up most of the information on the Internet. A significant part of this content are the videos superimposed with subtitles. Subtitles come in two varieties - external and embedded. External ones are contained in a separate file and are added by a video player software during playback. These subtitles can be easily replaced, edited or deleted if necessary, by appropriate actions with the subtitle file. Embedded subtitles are always displayed during video playback and are part of the video file. Thus, the task of localization and recognition of embedded subtitles is a task of image processing and can be solved by various relevant methods.

Embedded subtitles are text overlaid on video frames. At the same time, various text characters related to the displayed natural scene may also be present on the frames themselves. Thus, the text information contained in the frames of the video is divided into 2 types: the text located on the scene (Scene Text) and the text embedded from the outside source (Embedded Text). Subtitles are Embedded Text.

The Scene Text comes in great variety and variability. E.g. different frames of the video can have it written in several languages, being of different colors,

© Springer Nature Switzerland AG 2020
O. Gervasi et al. (Eds.): ICCSA 2020, LNCS 12254, pp. 119–128, 2020.
https://doi.org/10.1007/978-3-030-58817-5_10

fonts, sizes, orientations, and shapes. Quality of such text in the image cannot be guaranteed due to the shooting conditions and corresponding distortions—defocusing, poor lighting, shadows, blips, etc. In contrast, the Embedded Text on almost all frames of the video is presented in one language, normally is of the same size, font, and orientation. Image quality of the text is approximately the same for all frames—it has a high contrast and is well focused. In addition, the embedded text has temporal homogeneity—the frequency of text changes does not exceed a certain value, and for a certain period the text itself does not change nor does its position or orientation. These features make it possible to distinguish subtitles from a complex background, including the Scene Text, and to localize the position of subtitles by simpler methods than the ones used to detect and recognize text in the scene.

Traditional methods of localizing and recognizing text on images of video frames use one of three approaches. The first approach uses various classifiers from the area of sliding multi-scale windows. The second approach is based on applying classifiers to the static regions identified in the frame images. The third approach involves texture analysis of images.

In the first approach, each sliding window is tested for being part of a text segment. All windows classified as ones containing text are grouped into text regions according to certain rules. Then, one of the OCR classifiers is applied to each text region [1–3].

The second approach is much like the first but differs in the stage of finding a text region. In this case, the image is searched for static regions, each of which is checked for belonging to a text segment. Typically, in a static region, areas of uniform color or brightness are searched for by connected components analysis and then grouped together [4–6]. Connected areas that conform to certain constraints, such as size and shape, are then used as a basis for text extraction.

Texture-based analysis reveals periodic components in brightness or color, as well as in directions of the structures of text areas [7–9]. For texture features, one can use the results of applying various spectral transformations to the image. These transformations include Fourier transform, discrete cosine transform, and discrete multilevel wavelet transform. Texture analysis can be applied to the entire frame, as well as to sliding windows or static regions.

Recently, methods of deep learning have been in use to solve problems of localization and text recognition. In this case, text areas are treated as objects for segmentation. A pre-trained convolutional neural network is used to obtain a map of object segmentation. This map shows whether each pixel in the image belongs to a symbol, to a text area, or to another object [10,11]. Interconnected pixels of this map are marked as text candidates and are further detected as a character or as a text region. Many works [12,13] describe various architectures of convolutional neural networks that allow localization of individual characters and text regions, as well as text recognition in identified regions. It should be noted that great successes have been achieved in this approach. However, use of deep learning methods requires large training sets and sufficiently powerful computers; in addition, these methods cannot always work in real time. Therefore, using

these methods for localization and recognition of subtitles is quite expensive in terms of performance and computing power.

In this work, we propose a method for localizing subtitles in a video stream which is based on the mentioned features of the embedded text. The method is based on the analysis of static regions of image frames, it does not require large computational costs and can work in real time.

2 The Method of Detection and Localization of Subtitles in a Video Stream

The essence of the method is the search for static regions in the frame images with a subsequent analysis of connected areas within these regions in order to determine whether the region belongs to the subtitle area. To that end, a background model is built over several frames of the video. Based on this model, a binary background image mask is generated. An analysis of connected components reveals static regions in this mask, some of which may include subtitles. Then, using heuristic rules, an analysis is made of the relationship of parameters of static regions, which reveals the affiliation of these region groups to subtitles. The structure of the method for detecting and localizing subtitles is shown in Fig. 1.

2.1 Preliminary Processing

At this stage, frames images are extracted from the video stream one by one. Each image of the frame, depending on the resolution of the original video stream, may decrease by 2 or 4 times. This is required in order to reduce frame processing time. The standard background subtraction procedure is applied to the resulting frames. For separating background and foreground objects, we propose to use a method based on a Gaussian mixture model (GMM). This model is based on the use of information about the color change of an image pixel on several frames [14]. Since the text in the subtitles does not change for several frames in a row, the subtitles belong to the background. Applying this method yields two binary frame image masks, which constitute the result of preprocessing. The second mask can be used to detect changes in the position of the subtitles or changes in the text itself. Before further processing, the resulting masks are subjected to a morphological image opening operation in order to eliminate possible high-frequency noise. Figure 2 shows the video frame and its background mask which has been obtained from 5 preceding frames.

2.2 Static Area Segmentation

First, the background mask is binarized using the adaptive threshold [15]. For static regions segmentation, the connected components search method is applied to the binary background mask. The result is a grayscale image in which the brightness of each pixel corresponds to the number of the static region to which

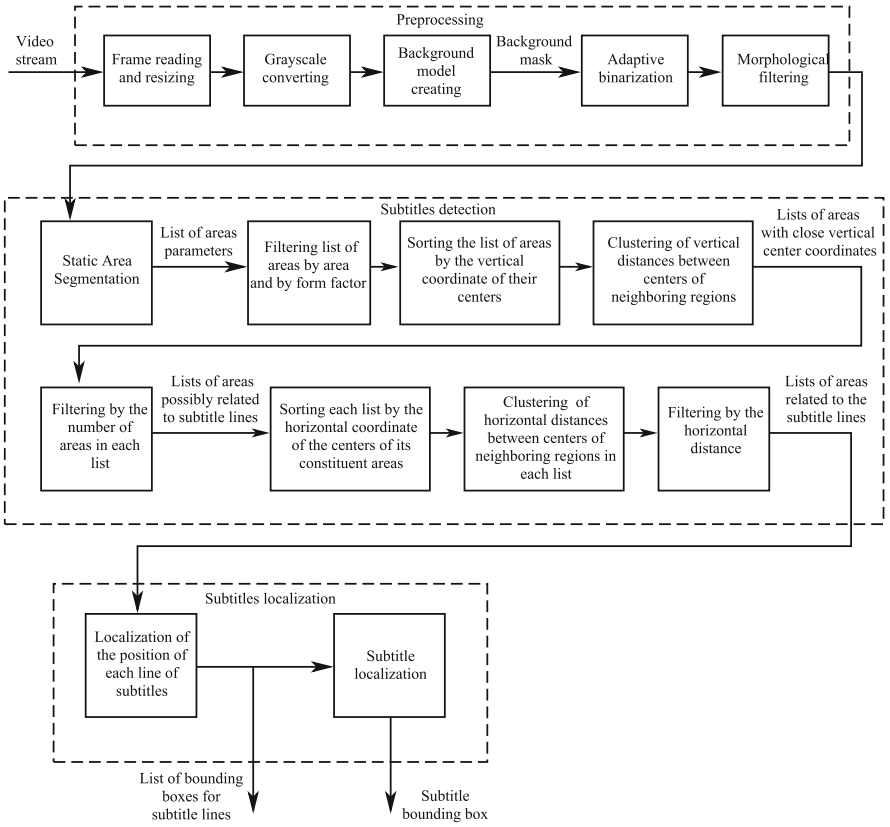


Fig. 1. The structure of the method for detecting and localizing subtitles.



a) b)

Fig. 2. Preprocessing: a) the video frame; b) its background mask.

this pixel refers. In addition, segmentation process creates a list of these regions. Each interconnected region is described by a set of parameters. This set includes the coordinates of the center of the region, the area of the region, and the coordinates and sizes of the bounding rectangles. These parameters are then used for subtitle areas localization. Figure 3 shows the results of segmentation of static regions obtained using a binary background mask.



a) b)

Fig. 3. Segmentation of static regions: a) the binary background mask; b) the static regions.

2.3 Detection and Localization of Subtitle Regions

At the first step, the list of regions obtained during segmentation is filtered by occupied area. Since the regions with subtitle symbols have a relatively small area, only regions having an area that lies in a certain range are included in the filtered list. The upper bound of the range is equal to the maximum possible size of a subtitle character area, while the lower bound depends on the size of small details in the frame image and on the interference. Any text character has a form factor which is defined as a height-to-width ratio of that character. For characters of the most common fonts, this ratio lies in a certain limited range. As such, the list is additionally filtered by a predefined range of form factors. Figure 4 shows the results of filtering static regions by area and by form factor.

Subtitles are lines of text located in a certain close area of the frame. Changes in their position are rather rare compared to the frame rate. Normally, text lines of subtitles are oriented horizontally. Therefore, a sign of the possible presence of subtitles is the concentration of the identified areas (possibly related to text characters) in some lines of the frame. In other words, the text line of the subtitle is displayed as a set of areas whose centers have approximately the same vertical Y-coordinate. Moreover, the distribution of horizontal X-coordinates of these centers should demonstrate a certain periodicity.

At the second step of the algorithm, the filtered list of regions is sorted by the vertical Y-coordinate of the centers of these regions. In this sorted list, differences between adjacent vertical coordinates $\{\Delta y_i = y_{i+1} - y_i\}$ are calculated. The resulting set of these differences is divided into two classes using the clustering procedure. Relatively small differences go to one class, while relatively large ones



Fig. 4. The results of filtering static regions by area and by form factor. White dots indicate the location of the centers of the identified areas, presumably related to subtitles

go to the other class. The class with small differences supposedly includes areas with subtitle lines. The average value of this difference with some tolerance, say 2.5σ, sets the range of values for checking whether the area belongs to a text string in the vicinity of the vertical coordinate being checked. Next, the sorted list is scanned, with creating dedicated lists of areas having approximately the same vertical coordinates of their centers. Lists containing fewer elements than a predefined threshold, for example 5, are ignored in the subsequent steps of the algorithm. The number of lists remaining may likely match the number of text lines in subtitles. These lists are then checked to see whether the areas they contain belong to text lines of subtitles.

At the next step of the algorithm, we check whether the remaining lists of regions have any periodicity in the distances between the centers of the regions along the horizontal X-coordinate. In order to do this, like at the previous step, each list is sorted by the horizontal X-coordinate. Then we create a set of distances between the centers of regions in the horizontal direction $\{\Delta x_i = x_{i+1} - x_i\}$. A clustering procedure is also applied to the generated sets, which splits each set into two classes. Possible distances between characters are grouped in one class, and distances between words in a string are grouped in another class. Next, areas in each list are checked for being part of a subtitle line. If the range of distances in the first class is not too big, i.e., possible distances between the characters are approximately the same, then this list contains areas of the subtitle text string. In case of a large spread of distances, the assumption is that this list does not contain areas belonging to the subtitle line. If no periodicity is detected in any of these lists of regions, the assumption is again

that the given frame does not contain subtitles. Figure 5 shows the lines with periodicity in horizontal distances between the centers of the regions.

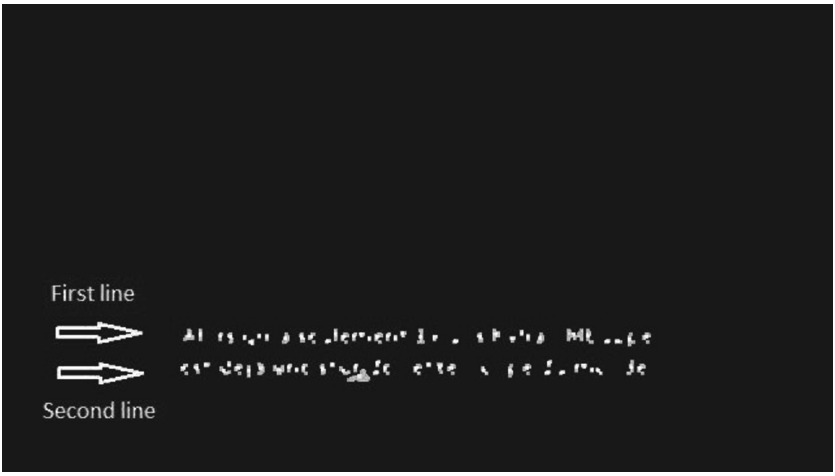


Fig. 5. The lines with periodicity in horizontal distances between the centers of the regions

After the lists of areas being part of the subtitle lines are identified, the search for the parameters of the bounding rectangles is performed, both for each detected text line and for the entire subtitle area. This search utilizes information about locations and sizes of local areas' bounding rectangles $\{x_i, y_i, w_i, h_i\}$ which can be found in lists corresponding to individual text lines of subtitles.

Here x_i, y_i are the coordinates of the upper left corner, and the w_i, h_i are the width and the height with a rectangle with index $i = 1, \ldots, N$, where N being the number of areas in the list. Figure 6 shows the areas and their bounding rectangles in the detected subtitle lines.

Since these lists have been ordered earlier by horizontal and vertical coordinates, the parameters X, Y, W, H of the bounding rectangle for the subtitle line are calculated as

$$\begin{aligned} X &= \min(x_i), \\ Y &= \min(y_i), \\ W &= \max(x_i) - \min(x_i) + w_{x_{max}}, \\ H &= \max(y_i) - \min(y_i) + w_{y_{max}}. \end{aligned}$$

Thus, each subtitle line is localized. The bounding rectangle for the entire subtitle area is defined in a similar way. Figure 7 shows bounding rectangles both for each detected subtitle line and for the entire subtitle area. The subtitle areas detected by the method are slightly smaller than the real areas, so we proportionally expand the calculated bounding rectangles by 5%. This allows you to more

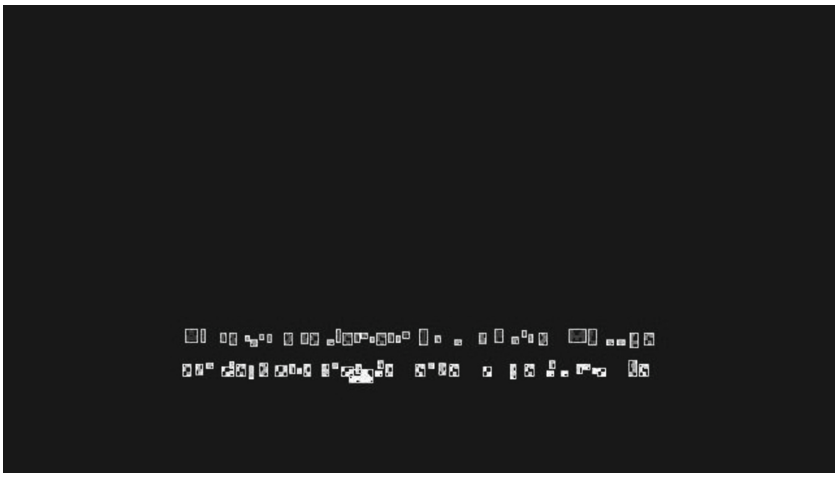
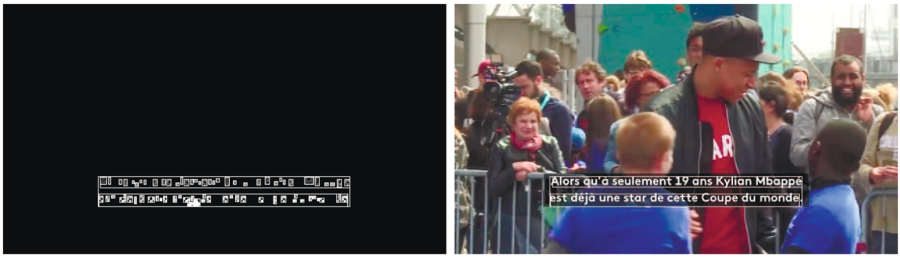


Fig. 6. The areas and their bounding rectangles in the detected subtitle lines.

accurately localize the area of subtitles. The localized image areas with subtitles are then passed on to the OCR system for recognition of the subtitle text.



a) b)

Fig. 7. The bounding rectangles for subtitle lines and for the entire subtitle area : a) in the binary mask; b) in the the video frame.

3 Experimental Results

The proposed method for detecting and localizing embedded subtitles is implemented in C# using the EmguCV computer vision library with support for GPU platform computing. An experimental verification of the method was carried out on several videos containing embedded subtitles. The subtitles in these videos were in different parts of the frame, came in different sizes and colors, however had enough contrast. The results of localization of such subtitles are shown in Fig. 8. This figure also shows the results of processing a frame without subtitles,



Fig. 8. The results of localization of subtitles in the video stream : a), b) true subtitles localization; c) no subtitles; d) false localization text as subtitles

as well as false detection of subtitles in the frame where a text document is displayed. The throughput of the implemented method on a 2.5 GHz Intel Core i5 processor is 20 FPS without GPU support and about 46 FPS when using GPU tools. The resulting processing speed, even without using a GPU, allows you to process the video stream and extract embedded subtitles in real time.

4 Conclusion

This paper proposes a method for detection and localization of embedded subtitles in a video stream. The method is based on the search for static regions in the frames of the video stream and on the subsequent analysis of the connected areas inside these regions. Experimental results show the effectiveness of the proposed method and prove the possibility of its application in real time. However, if the video stream contains a long streak of frames displaying text documents, the method may erroneously detect text strings of these documents as subtitles. This drawback of the method can be eliminated by counting the total number of lines in the identified subtitle area and filtering this value by a given threshold.

References

1. Keechul, J., Kwang, K., Anil, J.: Text information extraction in images and video: a survey. Pattern Recogn. **37**(5), 977–997 (2004). https://doi.org/10.1016/j.patcog.2003.10.012
2. Yin, X.-C., Zuo, Z.-Y., Tian, S., Liu, C.-L.: Text detection, tracking and recognition in video: a comprehensive survey. IEEE Trans. Image Process. **25**(6), 1–1 (2016). https://doi.org/10.1109/TIP.2016.2554321
3. Adam, C., et al.: Text detection and character recognition in scene images with unsupervised feature learning. In: 2011 International Conference on Document Analysis and Recognition (ICDAR), pp. 440–445. IEEE (2011). https://doi.org/10.1109/ICDAR.2011.95
4. Daehyun, K., Kwanghoon, S.: Static text region detection in video sequences using color and orientation consistencies. In: 19th International Conference on Pattern Recognition, ICPR 2008, pp. 1–4 (2008). https://doi.org/10.1109/ICPR.2008.4761629
5. Satish, H., Suresh, V.: Character recognition of video subtitles. ICTACT J. Image Video Process. **07**(02), 1351–1356 (2016). https://doi.org/10.21917/ijivp.2016.0196
6. Bahman, Z., Jingyue, C., Peter, W.: Instantaneously responsive subtitle localization and classification for TV applications. IEEE Trans. Consum. Electron. **57**(1), 274–282 (2011). https://doi.org/10.1109/TCE.2011.5735513
7. Felzenszwalb, P., Huttenlocher, D.: Pictorial structures for object recognition. Int. J. Comput. Vision **61**(1), 55–79 (2005). https://doi.org/10.1023/B:VISI.0000042934.15159.49
8. Mishra, A., Alahari, K., Jawahar, C.V.: Unsupervised refinement of color and stroke features for text binarization. Int. J. Doc. Anal. Recogn. (IJDAR) **20**(2), 105–121 (2017). https://doi.org/10.1007/s10032-017-0283-9
9. Zhou, X., et al.: EAST: an efficient and accurate scene text detector. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
10. Yao, C., Bai, X., Shi, B., Liu, W.: Strokelets: a learned multi-scale representation for scene text recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4042–4049 (2014)
11. Zhang, Z., Zhang, C., Shen, W., Yao, C., Liu, W., Bai, X.: Multi-oriented text detection with fully convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
12. Busta, M., Neumann, L., Matas, J.: Deep textspotter: an end-to-end trainable scene text localization and recognition framework. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2223–2231. IEEE (2017). https://doi.org/10.1109/ICCV.2017.242
13. Valery, G., Alexander, E., Oleg, I.: Localization of text in photorealistic images. In: Misra, S., et al. (eds.) ICCSA 2019. LNCS, vol. 11622, pp. 825–834. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-24305-0_63
14. Horprasert, T., Harwood, D., Davis, L.S.: A statistical approach for real-time robust background subtraction and shadow detection. In: IEEE ICCV, vol. 99, pp. 1–19 (1999)
15. Chan, F.H.Y., Lam, F.K., Zhu, H.: Adaptive thresholding by variational method. IEEE Trans. Image Process. **7**(3), 468–473 (1998)