Available online at: www.basra-science-journal.org

ISSN -1817 -2695

Arabic-text Extraction from Video Images

Abbas H. AL-Asadi and Thika Ali H. Subber
Computer Science Department, Science College, Basra University, Basrah, Iraq
Abbashh2002@yahoo.com

Received 13-8-2013 , Accepted 15-12-2013

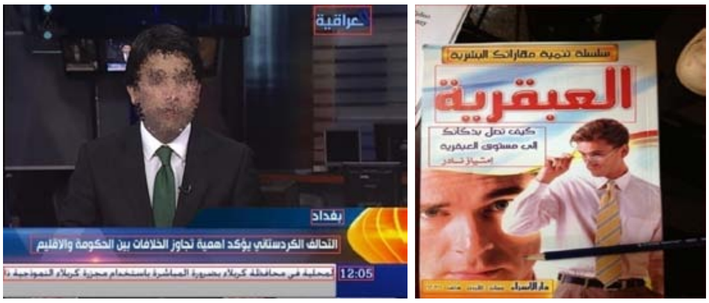
Abstract

Automatic extraction of meaningful objects in video is extremely useful in many practical applications. Text objects embedded in video contain much semantic information related to the multimedia content. In this paper, we proposed an algorithm to detect/localize and segment Arabic static artificial texts embedded in video. Firstly, the edge map of a predetermined region of interest is detected in a single video frame. Projection profiles are adopted to localize the candidate text regions followed by some filtering rules which are used to filter out non-text regions. Secondly, since artificial text lasts for a certain time on screen for reasons related to human vision, therefore the temporal information are exploited in order to reinforce the recall and precision rates. Finally, a thresholding-based method is used for separating text pixels from background pixels and produce a binary text image. To improve high recognition rate, robust enhancement methods are adopted pre/post text segmentation. Experimental results show that our algorithm is robust.

Keywords: Text Detection, Backward and Forward Text Tracing, Multi frame Integration

1. Introduction

Video is the most effective media for capturing the world around us. A large variety of video-based applications, such as video on demand, interactive TV, digital library, online distance learning, remote video surveillance, video conferencing, and so forth, has attracted much interest and attention. Efficient ways to analyze, annotate, browse, manipulate, and retrieve videos of interest based on their contents are becoming increasingly important and have attracted substantial research attention over the past decade accordingly [1]. Text objects embedded in video contain much semantic information related to the multimedia content.


عراقية
بغداد
التحالف الكردستاني يؤكد اهمية تجاوز الخلافات بين الحكومة والاقليم
12:05
العبقرية

(a) (b)

Figure 1. Artificial and Scene Texts (Red rectangles). (a) Artificial texts, and (b) Scene texts.

Text in video images can be classified into Artificial text and Scene text. **Artificial text** is artificially overlaid on the video image at the time of editing such as: newscast titles, films subtitles, sport scores, persons of interest names, etc. (See Figure 1.a). **Scene text** is naturally existed in the field of view of the camera during video capture such as: text on street signs, text on trucks, writings on shirts, etc. Its usefulness is confined to ad hoc applications such as: navigation and surveillance systems (See Figure 1.b). Artificial text is more descriptive and meaningful to the video content than the scene text.

In general, text embedded in video to be processed by computer easily should be structured (textural format). It is important and useful to be searchable, to be structured; it should be processed by optical character recognition (OCR). Unfortunately, direct feeding video image into OCR is irrational because from on one hand OCR is designed for scanned document image with completely clear background and on the other hand video image often has low resolution, color bleeding and complex background furthermore text in video image can be of different sizes, styles, alignments, and low contrast than background.

To overcome this case, text information extraction (TIE) problem can be divided into five sub-problems: (i) detection, (ii) localization, (iii) tracking, (iv) extraction and enhancement, and (v) recognition by OCR [2].

Text localization methods have been categorized into two types [2-4] : region-based and texture-based. *Region-based methods* use the properties of the color or gray-scale in a text region or their differences with the corresponding properties of the background. This category typically works in a bottom-up manner by separating the image into small regions and then merging these small regions into candidate or text regions based on several heuristics rules.

Region-based methods can be further subdivided into two types [2, 3]: Edge-Based methods and Connected Component-based methods. *Edge-based methods* [5-10] focus on the high contrast between the text and the background. In general, text is a set of connected components that appear in clusters at a limited distance and aligned to a horizontal or vertical line, since that is the natural method of writing words, *Connected Component-based methods* [3, 9-13] consider this characteristic.

Texture-based methods use the observation that the text in the images has distinct textural properties that distinguish them from the background .The techniques based on Gabor filters[14], Wavelet [15,16] , FFT, spatial variance, etc. can be used to detect the textural properties of a text region in an image[2]. Texture-based methods typically work in a top down way by extracting texture features of image followed by a classification stage then locating text regions. Texture-based methods are computation expensive specially in the classification stage [10] and it may confuse when text-like regions appear [9,10].

Text region segmentation methods are divided into three classes[8,17]:

First class(Threshold-based methods)[18,19]: Methods use either global or local or multilevel thresholds to retrieve text region (text pixels). *Second class(Stroke-based methods)*[20]: Methods use some filters to enhance stroke-like shapes and then detect strokes according to their density in order to retrieve text region. The features utilized by such filters are used to enhance the contrast at edges that are most likely to represent text. *Third class(Color-based methods)*[10]: Methods exploit the color information as they assume that a text pixels possesses a uniform color which is different from background pixels and by simply performing color similarity using clustering methods to divide text region into several color clusters, then conducts connected component analysis (CCA) to detect text components. In this paper, we proposed an algorithm to detect/localize, track and segment Arabic static artificial texts embedded in video images, (Figure 2) illustrates the steps of the proposed algorithm.

2. Arabic Language Overview

Arabic language has a very rich vocabulary. More than 422 million people speak this language as their native speaking language, and over 1 billion people use it in several religion-related activities , making it one of the half dozen most popular languages in the world. Although Arabic texts have vast popularity, they have not received enough interest by researchers, especially in the field of text extraction from video images. Arabic alphabet is represented numerically by a standard communication interchange code approved by the Arab Standard of Metrology Organization (ASMO) which resembles the American Standard Code for Information Interchange (ASCII). Each character in the ASMO code is represented by one byte [21].

The main characteristics of Arabic text can be summarized as follows:

1. A character may have several strokes.
2. Arabic characters consist of strokes with horizontal, vertical and diagonal directions.
3. Arabic letters can have more than one shape according to their position in the word: initial, middle, final, or standalone, Unlike English letters which have two shapes uppercase and lowercase.
4. In English, only two letters "i" and "j" in the lowercase representation have secondary character (complementary) that is located above the primary character which is called "dot or point", while in Arabic 15 letters out of 28 have dots which are located above, below or in the middle of the primary characters and 4 letters have another type of secondary character (complementary) that is located above or below the primary character which is called "Hamza".
5. Arabic characters are normally connected to each other to form meaningful word meaning that there are no gaps between most letters in a single word. Hence baseline of Arabic language is rich in pixels more than what is in baseline of English language.
6. Arabic language is horizontally alignment (i.e.: Arabic script is written and read from right to left).
7. Arabic letters are rich with angles or corners.

3. Candidate Text Detection/Localization

3.1 Text Region of Interest Determination

To produce a perfect Machine Vision System (MVS) we must emulate Human Vision System (HVS) accordingly. One can notice that each region in TV screen preserves for previewing type of information. By exploiting this fact we can determine the most descriptive text region related to the video being displayed, (Figure 3) shows text regions of our TV panel. As shown in (Figure 3), captions that appear in the *lower third portion* of a frame are almost used to describe a location, person of interest, title, subtitle, or event in news video. It is our optimal region of interest in our framework. A ticker tape is widely used in broadcasting news to display information such as the weather, sports scores, or the stock market. In some broadcasting news, graphics such as weather forecasts are displayed in a ticker-tape format with the news logo in the *lower right corner* at full opacity

[22], this region is used also to display time and date.

Caption that appears in the *lower left corner* of a frame is almost used to display information as a ticker tape mode such as SMS service mobile number at full opacity. Graphics that appear in the *upper right corner* of a frame are almost used to describe the channel logo for Arabic language TV at full or partial opacity. Caption that appears in the *upper left corner* of a frame is always used to display information related to the program being displayed such as "Live" or "مباشر" at full opacity. Sometimes, it is used to display another channel logo which may be considered as the source of the video being displayed. In sport videos this region is used to display match score at full opacity. Very little channels which exploit the *upper middle region* between the mentioned upper opposite corners to display information .

3.2 Edge Extraction from Region of Interest

The characteristics (1, 2, 4, 5, 6 and 7) of Arabic language mentioned in (Section 2) provide more pixels in text regions than background regions. In order to highlight those pixels over smooth background we apply Laplacian of Gaussian detector.

Laplacian of Gaussian (LOG) is an edge detection operator in which Gaussian low-pass filter (LPF) smoothing is performed prior to the application of the Laplacian edge detector because Laplacian is extremely sensitive to noise [23]. Luminance component of a predetermined ROI is filtered by 5x5 Laplacian of Gaussian mask with Sigma σ 0.52. Laplacian generates “double edges,” that is, positive and negative values for each edge.

In the flat regions and along the ramp, the Laplacian is zero. Large positive values of the Laplacian will occur in the transition region from the low plateau to the ramp; large negative values will be produced in the transition from the ramp to the high plateau [23] , in general speaking the transition in our case is the transition between text and background.

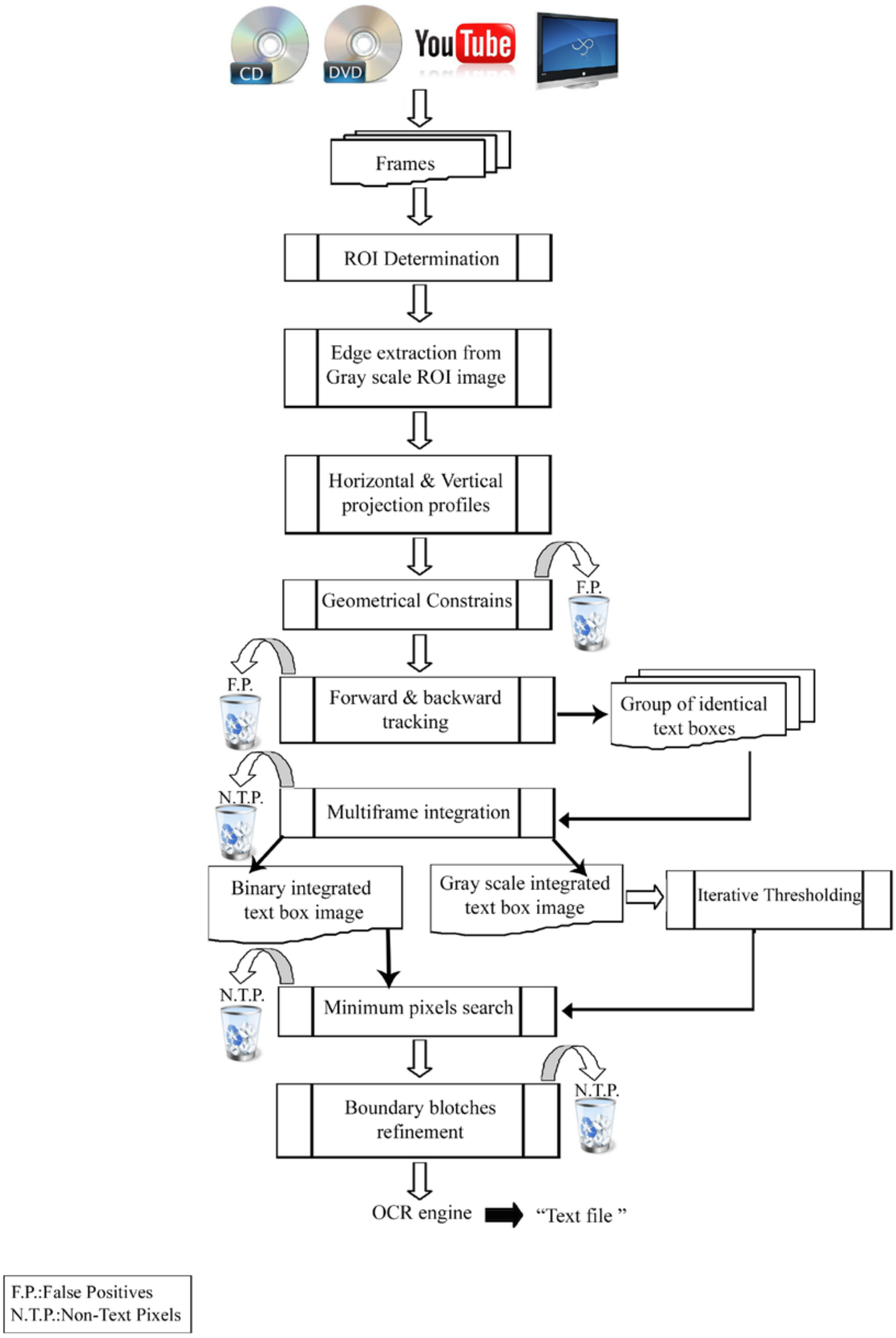

CD
DVD
You Tube
Frames
ROI Determination
Edge extraction from Gray scale ROI image
Horizontal & Vertical projection profiles
Geometrical Constrains
F.P.
Forward & backward tracking
F.P.
Group of identical text boxes
N.T.P.
Multiframe integration
Binary integrated text box image
Gray scale integrated text box image
Iterative Thresholding
N.T.P.
Minimum pixels search
Boundary blotches refinement
N.T.P.
OCR engine
"Text file "
F.P.:False Positives
N.T.P.:Non-Text Pixels

Figure 2. Steps of the proposed algorithm.

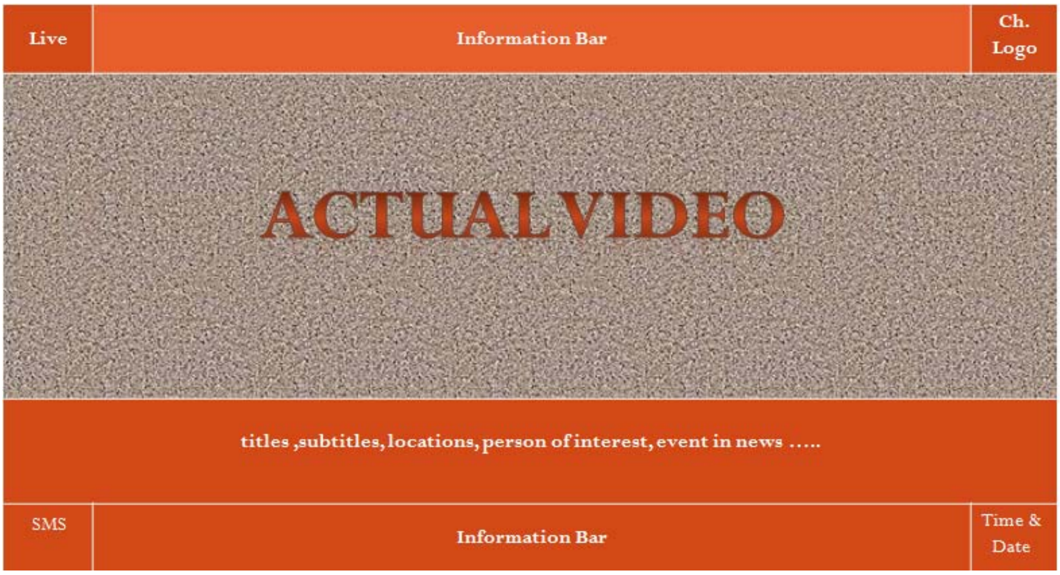


Figure 3. TV Text regions.

The Gaussian filter is defined as:

$$G(x,y)=\ell\left[-\frac{x^2+y^2}{2\sigma^2}\right] \text{….. Eq. (1)}$$

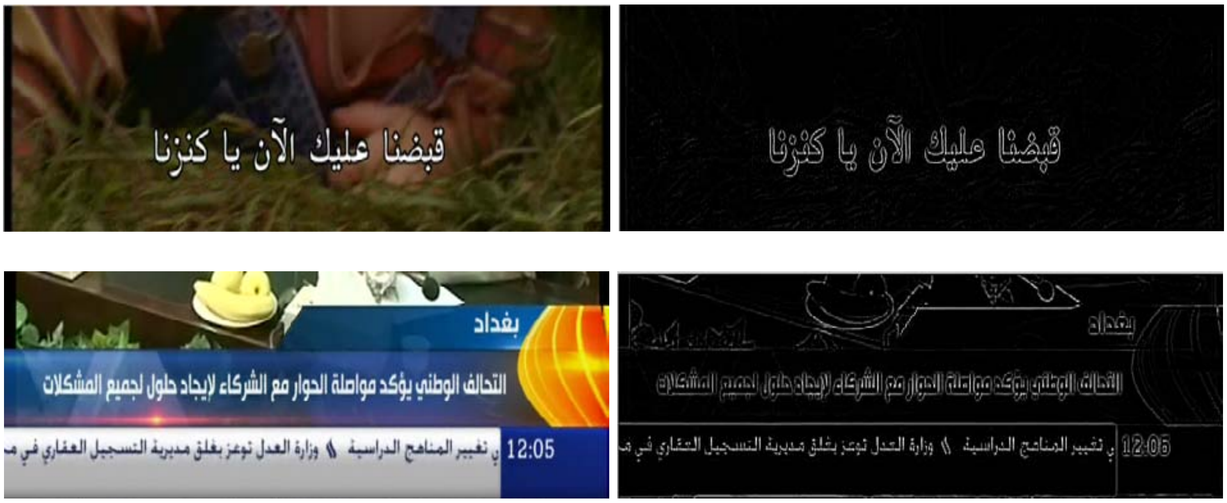
The Laplacian of an image $f\,(x, y)$ is defined as:

$$\nabla^2(x,y)=\frac{\partial^2(x,y)}{\partial x^2}+\frac{\partial^2(x,y)}{\partial y^2} \quad \text{….. Eq.(2)}$$

Therefore, Laplacian of Gaussian filter can be defined as:

$$h(x,y)=\nabla^2 G(x,y)=\frac{x^2+y^2-2\sigma^2}{\sigma^4}\,\ell\left(-\frac{x^2+y^2}{2\sigma^2}\right) \text{….. Eq. (3).}$$

(Figure 4) shows two examples of using Laplacian of Gaussian filter, one can notice that LOG operator is efficient for revealing the objects with high gradient magnitudes and discards others.


قبضنا عليك الآن يا كنزنا
بغداد
التحالف الوطني يؤكد مواصلة الحوار مع الشركاء لإيجاد حلول لجميع المشكلات
تغيير المناهج الدراسية وزارة العدل توعز بغلق مديرية التسجيل العقاري في م
12:05

(a) (b)

Figure 4. Two examples of using LOG. (a) original true color ROI images, (b) LOG filtered images.

3.3 Candidate Text Region Boundary Localization

Projections are very useful and compact shape descriptors [24], they belong to amplitude segmentation methods. Text region segments can be effectively isolated by forming the average *amplitude projections* of an image along its rows and columns. Before performing *amplitude projections*, we binarized the pre-LOG filtered image with an experimental threshold (0.44). Text strokes with low opacity may be affected with such global threshold (i.e. broken edges), therefore to overcome this problem and ensure that all text strokes will be included in the bounding box ,we connected these broken edges by using bridge morphological operation.

The horizontal and vertical projections of an edge map are defined by equations:

$$Hp(x)=\sum_{y=1}^{M} Bi(x,y) \quad \text{…. Eq. (4)}$$

Horizontal threshold: $\left[\dfrac{average(Hp)+\min(Hp)}{2}\right]$.

Where image *Bi* denotes to the binarized LOG filtered ROI image and M denotes to the total number of columns in the ROI. If HP (x) is greater than a horizontal threshold then row (x) is a part of a candidate text line.

$$Vp(y)=\sum_{x=1}^{N} Bi(x,y) \quad \text{….. Eq.(5)}$$

Vertical threshold : $\left[\dfrac{average(Vp)+\min(Vp)}{2}\right]$.

Where N denotes to the total number of rows in the pre-horizontally projected region .If VP(y) is greater than a vertical threshold then column (y) is a part of a candidate text line.

From (Figure 5), one can notice clearly a big gap between text words and text lines because vertical and horizontal profiles, for columns and rows, between different words and two text lines respectively, have *Zero* values caused by the Laplacian operator which has considered those rows or columns (regions) as flat regions and as mentioned before, the Laplacian magnitude will be *Zero* in the flat regions.

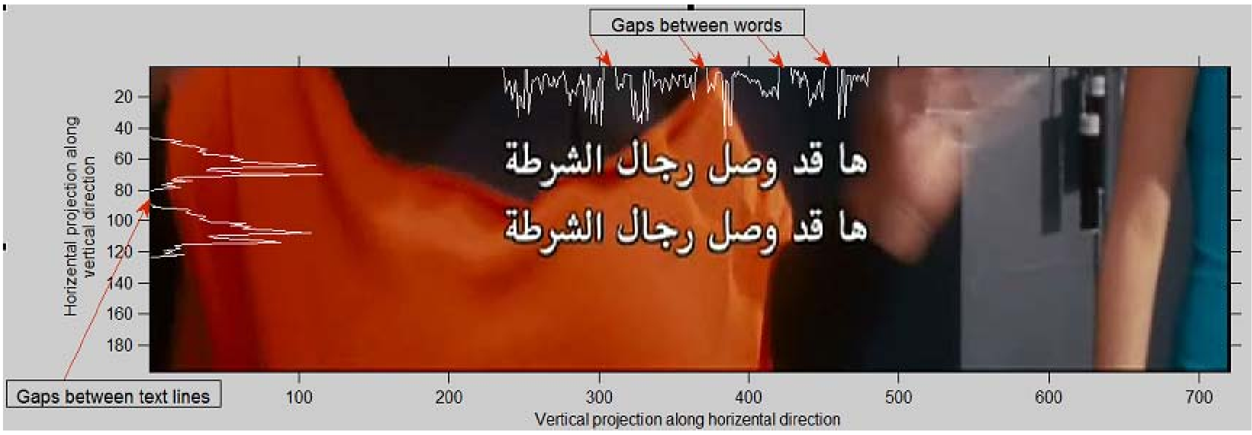


Figure 5. Horizontal and vertical projection profiles for a ROI image.

3.4 False Positive Elimination

False positives can be defined as falsely detected text regions (non-text regions). Based on the following geometrical rules which are related to Arabic characteristics, we can eliminate all or some of those false positives.

$$-\,Aspect\,ratio=\frac{Width}{Height}$$

$$-Area = Width \times Height$$

$$-Density = Mean = \frac{1}{NM}\sum_{x=1}^{N}\sum_{y=1}^{M} Bi(x,y)$$

-*Max Height* =Height of the region / 2.
-*Max Area* =Area of the region / 3.
The prelocated region must satisfy below rules to accept it as a candidate text region (See Figure 6).

1. Aspect ratio > threshold θ1.
2. Max Height >Height> threshold θ2.
3. Max Area >Area> threshold θ3.
4. Density > threshold θ4.

As Arabic characters have the mentioned characteristic (4 in Section 2), those small strokes often have low edge density, therefore their horizontal profiles will be below than a horizontal threshold. So to overcome this problem, we can extend the bounding box at the top and the bottom by 20% from the height of the bounding box.

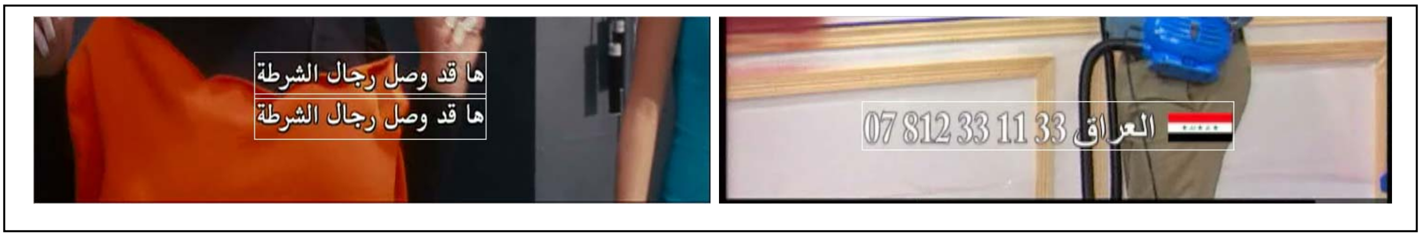


Figure 6. Accurate boundary of text regions.

4. Candidate Text Region Tracking

4.1 Forward and Backward Tracking

Once the candidate text region is detected, text tracking process is started. Text tracking can be defined as the following a pr-localized text event over time and determining the temporal and spatial information of text event. To enhance the system performance, it is necessary to consider temporal changes in a frame sequence. Text tracking process is interested to :

1. Verify the text localization results.
2. Speed up the overall system, if text tracking could be performed in a shorter time than text detection and localization, because text detection for every frame is computationally expensive and not necessary since the same text region lasts for a certain time [2].
3. Recover the original text image in cases where text is occluded in different frames.
4. Get the temporal range (initiative frame and terminative frame) of each text line.
5. Reduce the amount of storage in the database by emitting the duplicates.

Text tracking methods are based on the concept of the difference or similarity between two text regions from two consecutive frames. There are a lot of matching techniques for this purpose. In our experiments, we used absolute difference method because of its simplicity, fastness, and efficiency. Absolute difference can be defined as:

$$D(x,y) = \sum_{x=1}^{N}\sum_{y=1}^{M}\left|CTR_i(x,y) - CTR_{i+1}(x,y)\right| \quad \text{….. Eq.(6)}$$

Where CTR_i *(x, y)* denotes to the pixel value of the candidate text region in the LOG filtered image of frame $_i$, CTR_{i+1} *(x, y)* denotes to the pixel value of the candidate text region in the LOG filtered image of frame$_{i+1}$ and D is the image difference between the above two regions. Then we computed the ratio between Ones (white pixels)to the total pixels of the binarized difference image. If the ratio is less than a predetermined threshold θ5, then both regions are identical, otherwise, they are not. (Figure 7) illustrates the result of text tracking after using absolute difference.

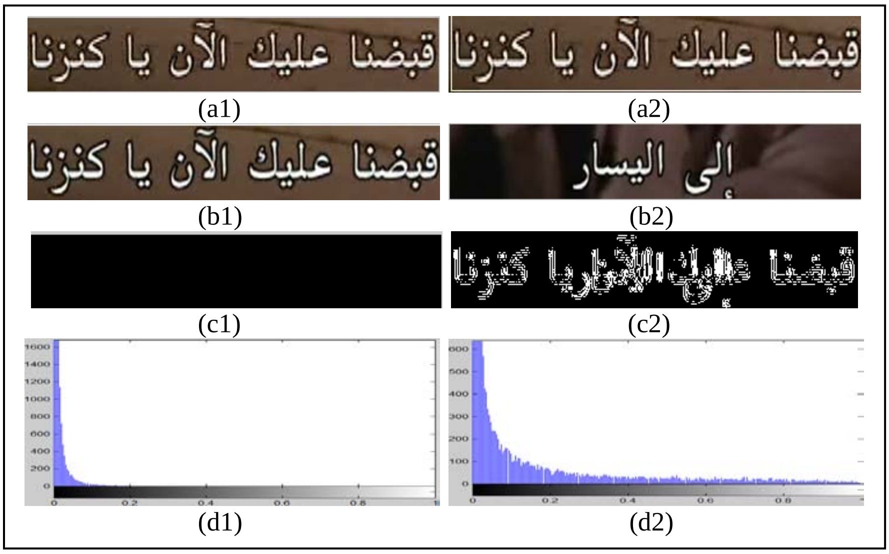


Figure 7. Text Tracking. In the first column, (a1&b1) are text boxes in two consecutive frames with similar text line. (c1) is the binarized image difference. (d1) is the histogram of image difference. The second column is the same as the first column but with different text lines.

We tracked each text box in both sides forward direction until it reaches its terminative frame and backward direction until it reaches its initiative frame. Lastly, we kept the temporal information (i.e.: the initiative frame and the terminative frame) and spatial information (i.e.: position) of each candidate text box. It is observed that, sometimes there is a text object that lasts for long period on TV screen, therefore, the tracking process will track that text object for long time (too many frames) and at the same time it ignores the possibility of the appearance of another candidate text object in another region in our ROI; therefore, we sampled the process of detection / localization stage every predefined threshold θ6 as shown in (Figure 8).

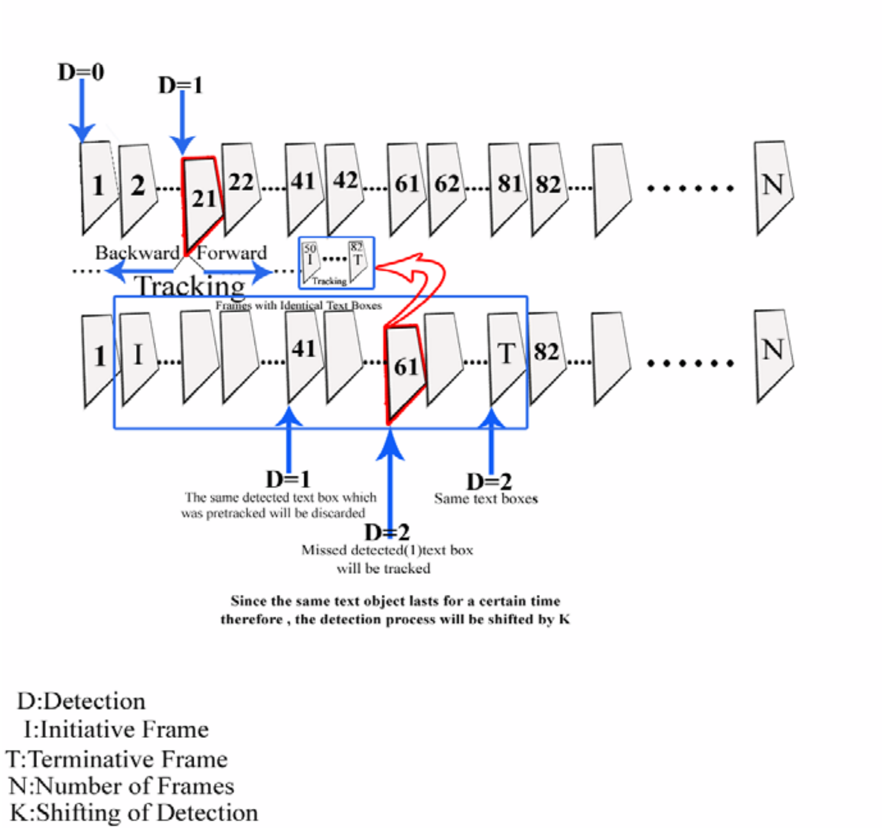

D=0
D=1
1
2
21
22
41
42
61
62
81
82
N
Backward
Forward
Tracking
50 I
82 T
Tracking
Frames with Identical Text Boxes
1
I
41
61
T
82
N
D=1
The same detected text box which was pretracked will be discarded
D=2
Same text boxes
D=2
Missed detected(1)text box will be tracked
Since the same text object lasts for a certain time therefore , the detection process will be shifted by K
D:Detection
I:Initiative Frame
T:Terminative Frame
N:Number of Frames
K:Shifting of Detection

Figure 8. Detection and Tracking Process .

But how can we know that the recently detected candidate text region is indeed missed detected or not ? We can do by computing the region location similarity between the recently detected candidate text region and all pre-tracked text regions which their terminative frames are still active. Region location similarity is proved by the overlap of two text regions from different frames. It includes Region area overlapping and Region position similarity (i.e.: (x_1, y_1)). Region area overlapping can be defined as :

$$Overlap = \frac{RA_i \cap RA_{i+1}}{Max(RA_i, RA_{i+1})} \quad \text{….. Eq.(7)}$$

If *overlap* is larger than a predetermined threshold *θ7* then the recently detected candidate text region was pretracked, otherwise it is missed.

4.2 False positive Elimination for Second Time

Since the text lasts at least 1 second on screen to be readable from human; therefore, we remove any candidate text object that lasts for less than a suitable threshold θ8. All candidate text regions which pass this false positive removal rule will be considered as texts.

5. Text Region Enhancement

Multiframe Integration

As a result of text region tracking process, we got two types of information: temporal information (i.e.: the initiative frame and the terminative frame) and spatial information (i.e.: position) of text line. By exploiting that information we can get a single enhanced text box image. Multiframe enhancement approaches [9, 20, 25, 26] are perfect for text region enhancement. Two types of integration are used:

A) Multi frame Gray level Integration

Since the color of pixels belonging to text strokes are identical over frames, while the color of background pixels are changing over frames, by utilizing this characteristic, gray scale integration can be used to improve the contrast between text and background in hand and get the approximated homogeneity of background in other hand which is (the background homogeneity) the crucial thing to the segmentation (binarization) step.

We adopted Min/Max pixel search method for multi frame gray level integration (enhancement).

It is observed that most captions have high intensity values against background such as white or yellow pixels, therefore, we can get the minimum value of the corresponding pixels from the *identical text boxes* (ITBs) of consecutive frames. The integration can be described as:

$$GI(x,y) = \{Min(g_k(x,y),...,g_{k+N}(x,y))\} \forall k : k = 1..N \quad \text{….. Eq.(8)}$$

Where *GI(x,y)* denotes to the pixel value of the integratd Gray scale text box image , $g_k(x,y)$ denotes to the pixel value of the *K* gray scale text box image and *N* denote to the size of ITBs (*total number of identical text boxes*). It is shown in (Figures 9.d and 10.b) that background becomes unclear, bluring and slightly homogenous (nearly to dark) while text becomes clearer against background.

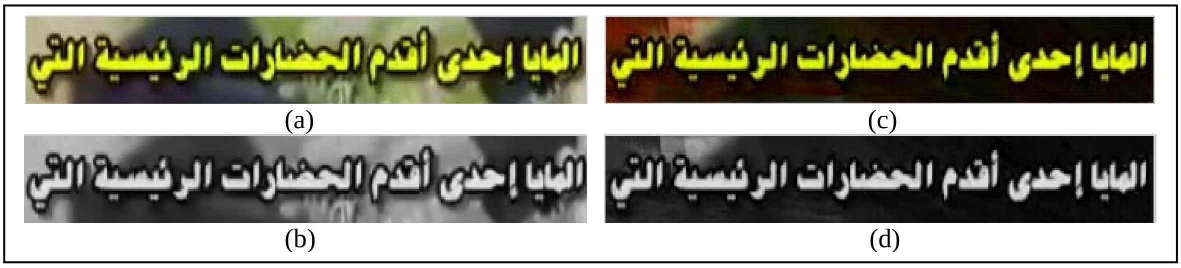
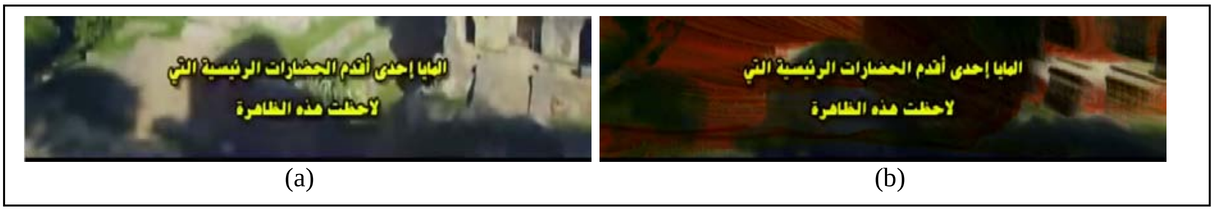


Figure 9. MFI/Gray level integration. (a & b) true color & gray scale text box image respectively. (c & d) text boxes after applying multi frame integration to (a & b) respectively.



(a) (b)

Figure 10. MFI/Gray level integration. (a) true color ROI image. (b) after applying multi frame integration to (*a*).

B) Multi Frame Binary level Integration

Since static artificial text object has the same position over multiple frame, using this characteristic can get an integrated binary text box image from the *identical text boxes* (ITBs) of consecutive frames where each pixel value of an integrated binary text box image denotes to its immortality over frames. Congjie Mi *et al.* [25] have benefits from edge information over multiple frames. We have benefits from binary text box integration over multiple frame .The binary level integration can be described as follows:

$$I\ (x,y)=\sum_{k=1}^{k+N} Bg_k(x,y), \forall k : k=1..N \text{ …. Eq. (9).}$$

Where *I(x,y)* denotes to the pixel value of the integratd text box image (not binary) , *Bg*$_k$ *(x,y)* denotes to the pixel value of the *K* binary text box image and *N* denotes to the size of ITBs(total number of identical text boxes). *Bg*$_k$ is generated by using Otsu threshold selection method [27].

Then appearance (stability) possibility of each pixel is computed over mutiple frames as follows:

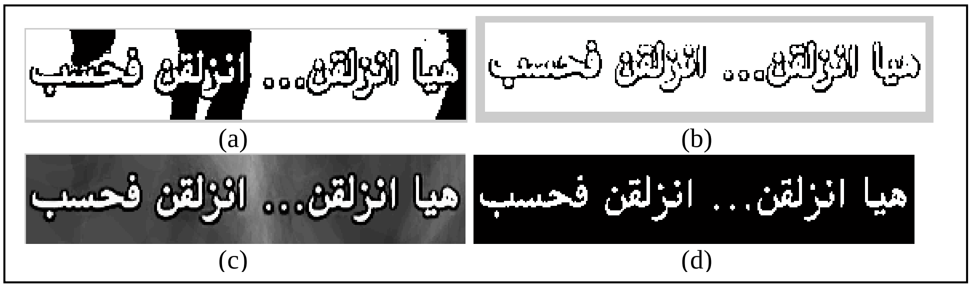
$$AP(x,y)=\frac{I\ (x,y)}{Size\ of\ ITRs} \text{ …. Eq. (10).}$$

Where *AP(x,y)* denotes to the appearance (stability) possibility of *I(x, y)*over mutiple frames.

If *AP(x,y)* is larger than a predefined threshold θ9 then it is static pixel over multiple frames and assures that it belongs to text pixels else if *AP(x,y)* is fewer than a predefined threshold θ9 then it is non-static pixel over multiple frames and assures that it belongs to background pixels.

$$BI(x,y)=\begin{cases}1 & if\ \ AP(x,y)>\theta 9\\ 0 & otherwise\end{cases} \text{ …. Eq. (11).}$$

Where *BI(x,y)* denotes to the pixel value of the integratd binary level text box image. (Figure 11) shows the result of binary level integration, one can notice that, multiframe binary level integration succeeded to remove the background complexity of (Figure 11.a)and generated one binary text box image with clear background as shown in (Figure 11.d).



(a) (b) (c) (d)

Figure11. MFI/ Binary level integration. (a) original binary text box image. (b) image *I* in Eq.9. (c) image *AP* in Eq.10 (d) image *BI* in Eq.11 (integrated binary text box image).

6. Text image Segmentation(Binarization)

Since most OCR systems already work with binary images whether normal or inverse binary image. Our proposed approach falls within the scope of these images. In general, text segmentation is used for splitting text pixels from background pixels and generates a homogenous regions based on a suitable criterion followed by binarization by means of marking text as one binary level and background as the other. What the segmentation step needs is the homogeneity. Our segmentation method is thresholding-based method because we already have the homogeneity of background caused by multi frame gray level integration step, (See Figure 12). Iterative selection method [28] is adopted to binarize the integratd gray scale text box as shown in (Figure 12.e).

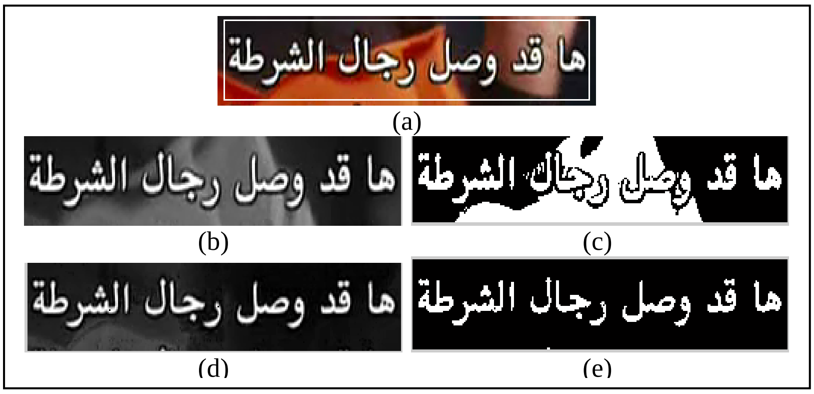


Figure 12. Text segmentation. (a)original true color text box image. (b) gray scale text box image. (c) binary image of *B*. (d) *b* after (MFI/Gray level integration). (e) binary image of *d*.

7. Enhancement of Segmentation

7.1 Binarized Text box image Enhancement

We can directly feed the binary text image (for example *e* in Figure 12 or *d* in Figure 11) to the OCR but unfortunately sometimes there are still disturbances overlaid on text image making OCR confuse to make its decision about a character, the reason belongs to the fewer changes at some regions of video event over frames. By making integration between the binary image which is generated from (Section 5.b) and the binary image which is generated from (Section 6), we can remove some of those disturbances as a preliminary enhancement step. The integration can be described as follows:

$$EBI(x,y) = \{Min(BI(x,y), BGI(x,y))\} \text{….. Eq.(12)}$$

Where *EBI(x,y)* denotes to the pixel value of the enhanced binary text box image, *BI(x,y)* denotes to the pixel value of the integratd binary level text box image and *BGI(x,y)* denotes to the pixel value of the binarized integratd gray scale text box image.

7.2 Boundary Blotches Refinement

As a secondary enhancement step and to improve high recognition rate we eliminate any blotches (non-text CCs) which often touches the boundary of text box image (See Figure 13).

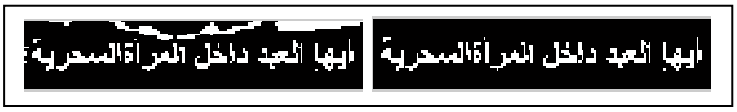


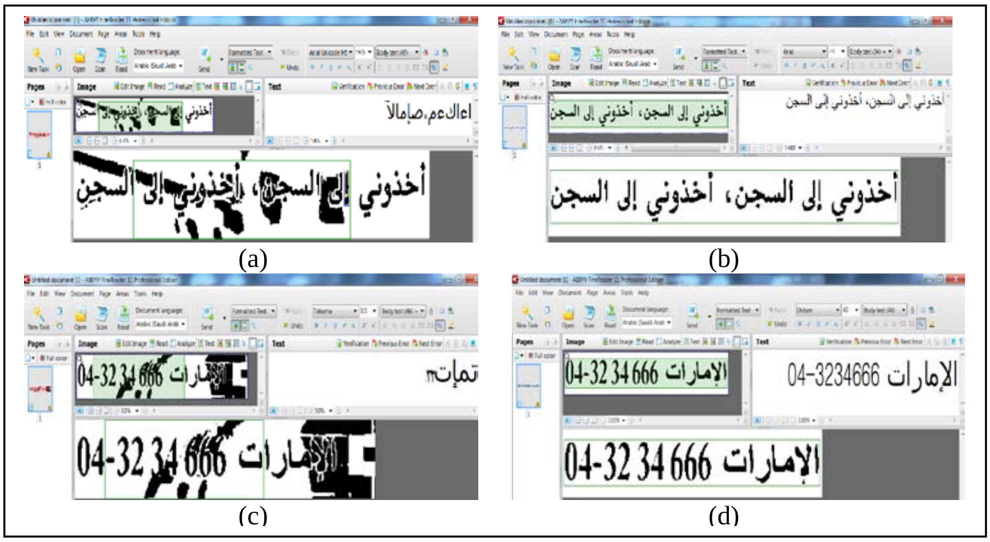
Figure 13. Boundary Blotches Refinement

8. Text Recognition

Finally, feeding the enhanced binary text image into OCR software. So, all previous stages actually work in complementary for doing OCR requirements because we don't have another technique for reading text image yet. Although text image is perfectly binarized as subjective measure but it gives today's standard OCR systems a hard time because these OCR systems have been designed to recognize text in documents, which were scanned at a resolution of at least 200–300 dpi, however text image have been recognized somehow correctly and we can use its text file simply.

There are two famous OCR software supported for Arabic languages : Abbey FineReader version 11[29] and Iris readiris V12 Middle East[30].

We have chosen Abbey FineReader because of its widespread application. Furthermore, more than 20 million people around the world use ABBYY FineReader at home and office for text recognition and document processing. In addition, ABBYY FineReader provides dictionary support for 36 languages (include of Arabic). This enables secondary analysis of the text elements on word level, with dictionary support; the program ensures even more accurate analysis and recognition of documents and simplifies further verification of recognition results [29] (See Figure 14).



(a) (b) (c) (d)

Figure 14. Arabic text image recognition by Abbey FineReader. (a)&(c) original binary text box images without any processing. (b)&(d) text box images after processing .

9. Experimental Results

The framework project work is designed in Matlab in 64 bit system with 2.5 GHZ core i5 processor where different videos are implemented for experiments. The graphics video display card of NIVIDIA GEforce is used with windows 7 operation system. Our experiments consider video with single text line, multiple text lines, text with different sizes, text with different fonts, text with complex background, and text with simple background. Since there is no common test video for Arabic caption extraction methods, we depend on our own dataset. We perform experiments on five different videos and the results were very satisfactory.

Performance of detection/localization step:

- **Detection Recall Rate :** indicated by the rate of correctly detected text boxes and the ground truth (total text boxes).

- **Detection Precision Rate:** the ratio of correctly detected text boxes to the sum of correctly detected text boxes plus false positives.

$$Detection\ Recall\ Rate = \frac{Number\ of\ correctly\ detected\ text\ boxes}{Ground\ truth\ (G.T)}$$

$$Detection\ Precision\ Rate = \frac{Number\ of\ correctly\ detected\ text\ boxes}{Number\ of\ correctly\ detected\ textboxes + false\ positives}$$

Performance of segmentation (binarazation) step:

Unfortunately, there is no objective evaluation for binarization step has presented yet, therefore the best way to measure a segmentation is to evaluate its OCR output. But, sometimes text line is correctly segmented as subjective measure (i.e.: still recognizable by human, has the readability factor and does not miss the main strokes) but not correctly recognized. Therefore, we can measure the performance of segmentation step subjectively based on our vision.

$$Segmentation\ Recall\ Rate = \frac{Number\ of\ correctly\ segmented\ characters(human\ vision)}{Total\ number\ of\ characters(G.T.)}$$

Performance of recognition step:

- **Recognition Recall Rate:** the ratio of correctly recognized characters to the total number of characters (ground truth).

- **Recognition Precision Rate :** the ratio of correctly recognized characters to the total number of recognized characters.

$$Recognition\ Recall\ Rate = \frac{Number\ of\ correctly\ recognized\ characters}{Total\ number\ of\ characters(G.T)}$$

$$Recognition\ Precision\ Rate = \frac{Number\ of\ correctly\ recognized\ characters}{Total\ number\ of\ recognized\ characters}$$

Table 1. Video dataset parameters

Elements	Video 1	Video2	Video3	Video4	Video5
Acquisition type	Broadcast	DVD	Broadcast	Broadband	Broadcast
Semantic meaning of Video	Diplomatic conversation	National geographic report	Commercials	Full length film	Newscast
Channel name	AL-MUSTAKILA	---------	Al Thuraya	---------	Iraqia TV
Total number of Frames	400	400	400	400	400
Resolution	640x576	720x540	720x576	320x240	720x576
Background complexity	Simple Bg.	Complex Bg.	Complex Bg.	Complex Bg.	Simple Bg.
# Textboxes without Multiframe integration	400	502	313	364	800
Number of characters without Multiframe integration	15019	12696	5634	9794	23600
# Textboxes with Multiframe integration	2	8	3	4	2
Number of characters with Multiframe integration	73	206	54	105	59

Table 2. Our framework parameters

Threshold	Value	Threshold	Value	Threshold	Value
θ1	0.8	θ5	0.04	θ9	0.9
θ2	11	θ6	20 frames		
θ3	300	θ7	0.8		
θ4	0.06	θ8	20 frames		

Table 3. Performance of detection/localization step without help of Mutiframe integration process (%)

Video #	Recall Rate	Precision Rate
Video 1	100	99.75
Video 2	100	96.72
Video 3	100	100
Video 4	100	100
Video 5	67.87	68.63
Average	93.57	93.02

Table 4. Performance of detection/localization step with the help of Multiframe integration process (%)

Video #	Recall Rate	Precision Rate
Video 1	100	100
Video 2	100	100
Video 3	100	100
Video 4	100	100
Video 5	100	100
Average	100	100

Table 5. Performance of segmentation step (%)

Video #	Recall Rate
Video 1	100
Video 2	100
Video 3	90.74
Video 4	100
Video 5	100
Average	98.14

Table 6. Performance of recognition step (%)

Video #	Recall Rate	Precision Rate
Video 1	87.67	91.78
Video 2	93.20	94.12
Video 3	87.04	90.38
Video 4	75.24	79.00
Video 5	71.19	76.36
Average	82.86	86.32

As a result from the performance of recognition step and the performance of segmentation step, 84.43 of the correctly segmented characters were also recognized correctly.

10. Conclusion

We proposed an algorithm to detect/localize and segment Arabic static artificial text from video images based on spatial information supported by characteristics of Arabic text lines and temporal information. We evaluated the proposed approach on a dataset gathered by our self. Experimental results show the following:

1. It is extremely useful in many practical applications to train MVS on the regions of interest in any video. By determining the text region of interest, we preserve the memory cost, process time cost, decrease the number of bugs as a result decreases the probability of irrelevant browsing or retrievals because it is irrational to extract all text objects either relevant or irrelevant from the whole frame.
2. Our proposed algorithm can successfully detect/localize Arabic text lines from video images with encouraging performance rates due to the efficiency of LOG operator to highlight text regions by filter out low gradient magnitudes from hand and rigid geometrical constrains to remove non text regions from other hand.
3. Friendly temporal information is extremely useful from many aspects beginning with its support to the localization step by removing more false positives which could not be caught by geometrical constrains, and ending with its support to the segmentation step by amazing enhancement to text region and represented with multiframe integration to facilitate the binarization step, and above all of this we have benefited from the mentioned services of tracking process.
4. Efficient enhancement for the binarized text box image.

References

[1] Zhang Y., "Advances in Image and Video Segmentation", *IRM Press*, ISBN 1-59140-753-2, 2006.

[2] Jung K., Kim K. I., and Jain A. K., " Text Information Extraction in Images and Video :A Survey", *Elsevier Ltd Pattern Recognition Society*, Vol. 37, pp. 977–997, 2004.

[3] Moradi M., Mozaffari S., and Orouji A. A., " Farsi/Arabic Text Extraction from Video Images by Corner Detection ", *IEEE Iranian Machine Vision and Image Processing*, pp. 1-6, 2010.

[4] Zhang J., "Extraction of Text Objects in Image and Video Documents", university of South Florida, *Part of the American Studies Commons and the Computer Sciences Commons*, 2012.

[5] Jianyong S., Xiling L., and Jun Z., "An Edge-Based Approach for Video Text Extraction", *IEEE International Conference on Computer Technology and Development*, pp. 331-335, 2009.

[6] Phan T. Q., Shivakumara P., and Tan C. L., "A Laplacian Method for Video Text Detection", *International Conference on Document Analysis and Recognition*, pp. 66-70, 10th 2009.

[7] Kaushik K.S., and Suresha D., "Automatic Text Extraction in Video Based on the Combined Corner Metric and Laplacian Filtering Technique", *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, Vol. 2(6), pp. 2119-2124, June 2013.

[8] Shivakumara P., Bhowmick S., Su B., Tan C. L., and Pal U., "A New Gradient Based Character Segmentation Method for Video Text Recognition", *IEEE International Conference on Document Analysis and Recognition*, pp. 126-130, 2011.

[9] Zhou J., Xu L., Xiao B., Dai R., and Si S., "A Robust System for Text Extraction in Video", *IEEE International Conference on Machine Vision*, pp.119-124, 2007.

[10] Song Y., Liu A., Pang L., Lin S., Zhang Y., and Tang S., "A Novel Image Text Extraction Method Based on K-Means Clustering", *Seventh IEEE/ACIS International Conference on Computer and Information Science*, pp.185-190, 2008.

[11] K N N. M. and Kumaraswamy Y S, "Robust Model for Text Extraction from Complex Video Inputs Based on Susan Contour Detection and Fuzzy C-Means Clustering", *IJCSI International Journal of Computer Science Issues*, Vol. 8(5), No. 3, pp. 225-234, September 2011.

[12] Zhao X., Lin K., Fu Y., Hu Y., Liu Y., and Huang T. S., "Text from Corners: A Novel Approach to Detect Text and Caption in Videos", *IEEE Transactions on Image Processing*, Vol. 20, No. 3., pp.790-799, March 2011.

[13] Shi S., Cheng T., Xiao S., and Lv X., "A Smart Approach for Text Detection, Localization and Extraction in Video Frames", *IEEE International Conference on Information Technology and computer Science*, pp.158-161, 2009.

[14] Weldon T. P., Higgins W. E., and Dunn D. F., "Efficient Gabor Filter Design for Texture Segmentation", *Elsevier Pattern Recognition*, Vol. 29(12), pp. 2005-2015, December 1996.

[15] Yea Q., Huangb Q., Gaoa W., and Zhao D., "Fast and Robust Text Detection in Images and Video Frames", *Elsevier Image and Vision Computing*, Vol. 23, pp. 565–576, 2005.

[16] Lee C., Chiang Y., Huang H., and Tsai C., "A Fast Caption Localization and Detection for News Videos", *IEEE Second International Conference on Innovative Computing, Information and Control*, 2007.

[17] Huang X., Ma H., and Zhang H., "A New Video Text Extraction Approach", *IEEE / ICME International Conference on Multimedia and Expo*, pp. 650-653, 2009.

[18] Gu L., "Text Detection and Extraction in MPEG Video Sequences", *In Proceedings of the International Workshop on Content-Based Multimedia Indexing CBMI '01, Brescia, Italy*, September, pp. 233-240, 2001.

[19] Ntirogiannis K., Gatos B., and Pratikakis I., "Binarization of Textual Content in Video Frames", *IEEE International Conference on Document Analysis and Recognition*, pp. 673-677, 2011.

[20] Li L., Li J., and Wang L., "An Integration Text Extraction Approach in Video Frame", *IEEE Proceedings of the Ninth International Conference on Machine Learning and Cybernetics*, Qingdao, pp. 2115-2120, 11-14 July 2010.

[21] Hassin A. H., Tang X., and Liu J., "Printed Arabic Character Recognition Using HMM" *Journal of Computer Science and Technology*, V19(4), pp. 538-543, 2004.

[22] Bovik A., "Hand Book of Image and Video Processing, *Academic Press*, ISBN 0-12-119790-5, Printed in Canada, 2000.

[23] Pratt W. K., "Digital Image Processing", *WILEY*, ISBN: 978-0-471-76777-0, Printed in the United States of America, 2007.

[24] Marques O., "Practical Image and Video Processing Using Matlab", *IEEE press, Wily*, ISBN 978-0-470-04815-3, 2011.

[25] Mi C., Xu Y., Lu H., and Xue X., "A Novel Video Text Extraction Approach Based on Multiple Frames", *IEEE/ICICS Fifth International Conference on Information,*

Communications and Signal Processing, pp. 678-682, 2005.

[26] Lienhart R., and Wernicke A., "Localizing and Segmenting Text in Images and Videos", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol.12, No.4, pp. 256-268, April, 2002.

[27] Otsu N., "A Threshold Selection Method from Gray-Level Histograms", *IEEE Transactions on System, Man, and Cybernetic*s, Vol. 9, pp. 62-66, 1979.

[28] Gonzalez R. C., Woods R. E., and Eddins S. L., "Digital Image Processing Using Matlab", *Pearson Prentice Hall*, 2003.

[29] Abbyy finereader 11OCR. Availible : http:// www.abbyy.com

[30] Readiris corporate 12 middle east OCR. Available : http://www.irislink.com

أستخراج النص العربي من صور الفيديو

المستخلص

يُعتبر الاستخراج التلقائي للكائنات ذات المعنى في الفيديو مفيد للغاية في العديد من التطبيقات العملية. حيث تحتوي الكائنات النصية المضمنة في الفيديو على كثير من المعلومات الدلالية ذات الصلة بمحتوى الوسائط المتعددة. في هذا البحث المسند، أقترحنا خوارزمية للكشف عن وتحديد موقع النصوص العربية الاصطناعية الثابتة المضمنة في الفيديو. أولاً، يتم الكشف عن خريطة الحافة لمنطقة مهمة محددة سلفا في إطار فيديوي واحد. يتم اعتماد ملامح الإسقاط في تحديد موقع المناطق النصية المرشحة تليها بعض قواعد الترشيح التي تستخدم لإستبعاد المناطق غير النصية. ثانياً ، طالما يستمر النص الاصطناعي لفترة زمنية معينة على الشاشة ولأسباب تتعلق برؤية الإنسان لذلك يتم أستغلال المعلومات الزمنية من أجل تعزيز معدلي الأسترجاع والدقة. أخيراً، أستُخدم الأسلوب القائم على التعتيب لفصل النقط الضوئية للنص من النقط الضوئية للخلفية و إنتاج صورة نص ثنائية. ولتحسين معدل تمييز عالي ، أعتُمدت أساليب تحسين قوية قبل و بعد تجزئة النص . تظهر النتائج التجريبية أن لدينا خوارزمية قوية.