IJICTR International Journal of Information & Communication Technology Research

Volume 7- Number 2- Spring 2015 (19-31)

Text Localization, Extraction and Inpainting in Color Images using Combined Structural and Textural Features

Mohammad Khodadadi Azadboni
Electrical and Electronic Engineering Department
Shahed University
Tehran, Iran
mohammad64kh@gmail.com

Alireza Behrad
Electrical and Electronic Engineering Department
Shahed University
Tehran, Iran
behrad@shahed.ac.ir

Received: July 21, 2014-Accepted: February 3, 2015

Abstract— **In this article, a new approach is proposed for text detection, extraction and inpainting in color images. The proposed algorithm includes three stages. In the first stage, several gradient based operators and image corners are utilized to localize text blocks. We use a new block split and merging algorithm to enhance the accuracy of text localization algorithm. An SVM based text verification algorithm is then employed with a new set of features to reject non-text blocks. To inpaint text areas, we cluster different colors in the text blocks using k-means clustering algorithm and estimate background and text colors. Then a color segmentation algorithm is employed to detect characters' pixels accurately. In the third stage, the proposed inpainting algorithm is applied to restore initial image contents. The inpainting algorithm is based on a matching algorithm that considers the priority for inpainting the pixels. Experimental results and a comparison of the results with those of other methods show the efficiency of the proposed algorithm.**

Keywords- text detection; text localization; image inpainting; structural and textural features; color segmentation.

I. INTRODUCTION

Nowadays, text detection and recognition in image and video frames is a widespread research area. Text subtitles in videos and images are used to give more information about the image or video. Text detection and recognition algorithms are used in different applications like video and image retrieval and OCR applications. Video or image inpainting aims to restore missing or defective part of an image or video by utilizing spatial and temporal information from the neighboring scenes.

In this paper, we aim to extract and inpaint text areas in color images or video frames. The proposed algorithm is useful for various image and video processing applications like image and video retrieval, automatic text inpainting in image and videos with subtitles and so forth. Since the inpainting algorithm employs the information of neighboring pixels for recovering original data, it is necessary to extract text areas more precisely. Therefore the proposed algorithm includes three different stages: 1- text localization which detects text blocks in the image, 2- text extraction which extracts character areas more precisely and 3- text inpainting, which uses spatial information in the neighboring scenes to recover the original image contents. The proposed algorithm is fully automatic and does not require any human user intervention. Several approaches have been proposed for the text localization in images. Existing approaches for the text localization can be roughly divided into three groups, including 1- color based approaches [1, 2], 2- gradient based method

[3, 4] and 3- methods based on textural and structural synthesis [5-24].

Color based approaches assume a predefined color for the text. Therefore, they lose their generality when the text color is not predetermined. These methods analyze connected components after color segmentation for the text localization [2].

Gradient based approaches utilize the high contrast of the text areas for text detection. These approaches generally employ the directional gradient of the image for text localization. Gradient based approaches can be utilized both in color and intensity images. These algorithms generally apply a threshold to the outputs of the gradient based operators and extract connected components. Consequently, these approaches are referred as connected component based algorithms as well.

In [25] combined edge and color information are used for text detection in natural images. The method also utilizes stroke width information to filter out non-text areas. Shekar et al. [26] employed the fusion of Discrete Wavelet Transform (DWT) and gradient difference for text localization in video frames. This approach first extracts keyframes of the video and apply multilevel DWT to obtain text-like image. Then the output of the DWT image is filtered using a Laplacian filter and consequently maximum gradient difference is used to localize text regions.

Methods based on the structural or statistical features mostly utilize text classifiers to distinguish between text and non-text areas. These features are generally utilized to train a classifier for text area detection. These methods are mostly used for the verification of the extracted text blocks and are rarely used for the text area localization directly. Different text classifiers may be used for text verification like neural networks [11, 23, 27] and SVM based approaches [8, 15].

In [28] an improved Maximally Stable Extremal Region (MSER) based method is used for text detection. The method uses a hierarchical approach for text region extraction. The experimental results show the effectiveness of this algorithm for detecting multilingual texts. In [29], an approach is proposed for text detection based on wavelet transform and angle projection boundary growing. In this algorithm, video frames are divided into blocks and probable text blocks are identified by using a combination of wavelet transform and median-moments with k-means clustering. Yin et al. [30] employed adaptive clustering algorithm for multi-orientation text detection in natural scene images. The method utilizes several sequential grouping steps comprising morphology-based grouping via single-link clustering, orientation-based grouping via divisive hierarchical clustering and projection-based grouping algorithms.

The output of the text localization algorithm is mostly text blocks. To inpaint the text areas more efficiently, it is necessary to extract the text characters more precisely. Different algorithms have been proposed for the character extraction in the text blocks. Document binarization is the mostly used method for the text extraction. Local thresholding algorithms like methods proposed by Niblack [31], Wolf [32], Sauvola and Pietikainen [33], Hao [34] and Wu [35] estimate different local thresholds for image pixels. Some algorithms like Sauvola and Pietikainen [33] and Wu [35] may also combine local and global thresholding algorithms.

Inpainting is an approach to restore the damaged areas in image by utilizing the neighboring scene information. The efficiency of the inpainting algorithm is determined based on the similarity between the restored and the original image. Several approaches for image inpainting have been proposed, which they can be classified into three groups:

1. Algorithms based on image interpolation, which mostly utilize Partial Differential Equation (PDE) or variational approaches [36-41].
2. Methods based on texture synthesis or block copying algorithms [42-46].
3. The combination of interpolation and texture based approaches [47-54].

The first group takes color or intensity information of the pixels around the damaged region and diffuses the information into the damaged regions gradually in several iterations. BSCB [37], Curvature-Driven Diffusion (CDD) [41] and Total Variation (TV) algorithms [38-40] which are generally based on PDE algorithm are categorized in this group. They work relatively well in filling thin damaged regions. However, in large regions or region with sharp edges, they result in blurry edges. Their efficiency is also highly dependent on the input parameters and the large number of steps leads to high computational cost.

In algorithms based on texture synthesis, the region around the damaged area is searched to find a texture similar to the texture of the damaged pixels. Then the damaged pixel is replaced with a pixel with a similar texture. Guo et al. [44] proposed a method based on structure feature replication and morphological erosion. The method has the advantage of restoring several pixels at each inpainting iteration.

The inpainting method of Elango and Murugesan [42] is based on Cellular Neural Network (CNN). The method uses a recursive approach for inpainting, which increases the computational burden of the algorithm.

In [45] an inpainting algorithm was employed to enhance the efficiency of image compression algorithm. In this approach, some regions are intentionally and automatically removed at the encoder and are restored naturally by image inpainting at the decoder to increase the compression rate of the proposed algorithm. The inpainting algorithm in the decoder side receives some necessary information from the encoder to help restoration.

The third group of inpainting algorithms combines interpolation with texture synthesis to increase the efficiency of the inpainting algorithm. These methods generally decompose the input image into different textures and structures and apply texture synthesis and interpolation methods in a combined manner. The methods generally suffer from the high computational burden.

The algorithm of [53] is based on patch propagation by inwardly propagating the image patches from the source region into the interior of the target region patch by patch. In each iteration of patch propagation, the algorithm is decomposed into two procedures, 1- patch selection and 2- patch inpainting. The method used structure sparsity for patch selection. They assumed that the selected patch on the fill-front is the sparse linear combination of the patches in the source region.

In [52] instead of replacing the gray value with that of the matching point, the gradients of half-points in the neighborhood of a damaged point and its matching point are used to estimate the gray value of the damaged point, directly.

The inpainting algorithm of [54] decomposes the image into the sum of two functions representing the underlying image structure and image textures respectively. Then the functions are separately reconstructed and finally are merged to build the inpainted image. The method suffers from image blurring and low processing rate.

In this paper, we propose a new algorithm for text localization, extraction and inpainting, which is based on our previous work [8]. The proposed text localization algorithm is based on the image gradient and image corners. We use a new block split and merging algorithm to increase the accuracy of text localization algorithm. We also employ an SVM based text verification algorithm with new sets of textural features to reject non-text blocks. To extract text characters, we estimate background and text colors in the candidate text blocks using a clustering algorithm. Then based on the text color histogram, text areas are determined precisely. We use a fast inpainting algorithm to fill character areas, which is based on texture synthesis and priority based on image structure. This algorithm is based on the combination of erosion operation, matching and the idea of repairing damaged pixels with priority based on image structure.

The paper is organized as follows: in section II, we discuss the general block scheme of the proposed algorithm. In section III, we present text localization algorithm. The proposed algorithm for extracting characters is described in section IV. Section V represents the inpainting algorithm for text area filling. Experimental results appear in section VI and we conclude the paper in section VII.

II. General Block Scheme of the Algorithm

Fig. 1 shows the general block scheme of the proposed algorithm. As shown in the figure, our algorithm comprises three stages including 1- text localization, 2- text extraction and 3- text inpainting.

The aim of the text localization algorithm is to detect text blocks in the image or video frames. The text localization algorithm utilizes image gradient and image corners to locate initial text blocks. We then use a new block split and merging algorithm to increase the accuracy of text localization algorithm. In the second stage of text localization algorithm an SVM based text verification algorithm is used. The aim of text verification algorithm is to reject some non-text blocks that are extracted in the first stage of text localization algorithm.

The second stage of the proposed algorithm is text extraction. The aim of the text extraction algorithm is to extract text pixels from the text blocks. The algorithm starts with the estimation of text and background colors in the blocks using k-means clustering algorithm. In this stage, several text colors may be estimated. To estimate the true text color, the initial text colors are compared with background colors and the color with maximum distance to background colors is selected as the text color.

The final stage of the proposed algorithm is the inpainting stage. This stage aims to remove text areas and recover original data of the image. The proposed inpainting algorithm is an iterative algorithm based on texture synthesis and matching and the priority of the damaged pixels. The algorithm first extracts border of a text region. Then pixels with higher intensity variation in the border are inpainted. The algorithm follows by inpainting remaining pixels in the border. The algorithm is iterated to inpaint all the pixels of the text area.

III. Text Localization

The aim of the text localization algorithm is to detect text blocks in the image or video frames. Text connectivity and the higher contrast of text pixels are the main properties that are used here for the extraction of the text blocks.

To increase the efficiency of the proposed algorithm for text localization, a multistage algorithm is employed. In the first stage, initial text locations are detected using a combined corner and contrast based operators. Then text blocks are determined using block splitting and merging algorithm, which merges or splits different text blocks based on their horizontal or vertical projection profiles. In the next step an SVM based text verification classifier is used to reject some non-text areas. Then, the block split and merge algorithm is repeated to refine text blocks.

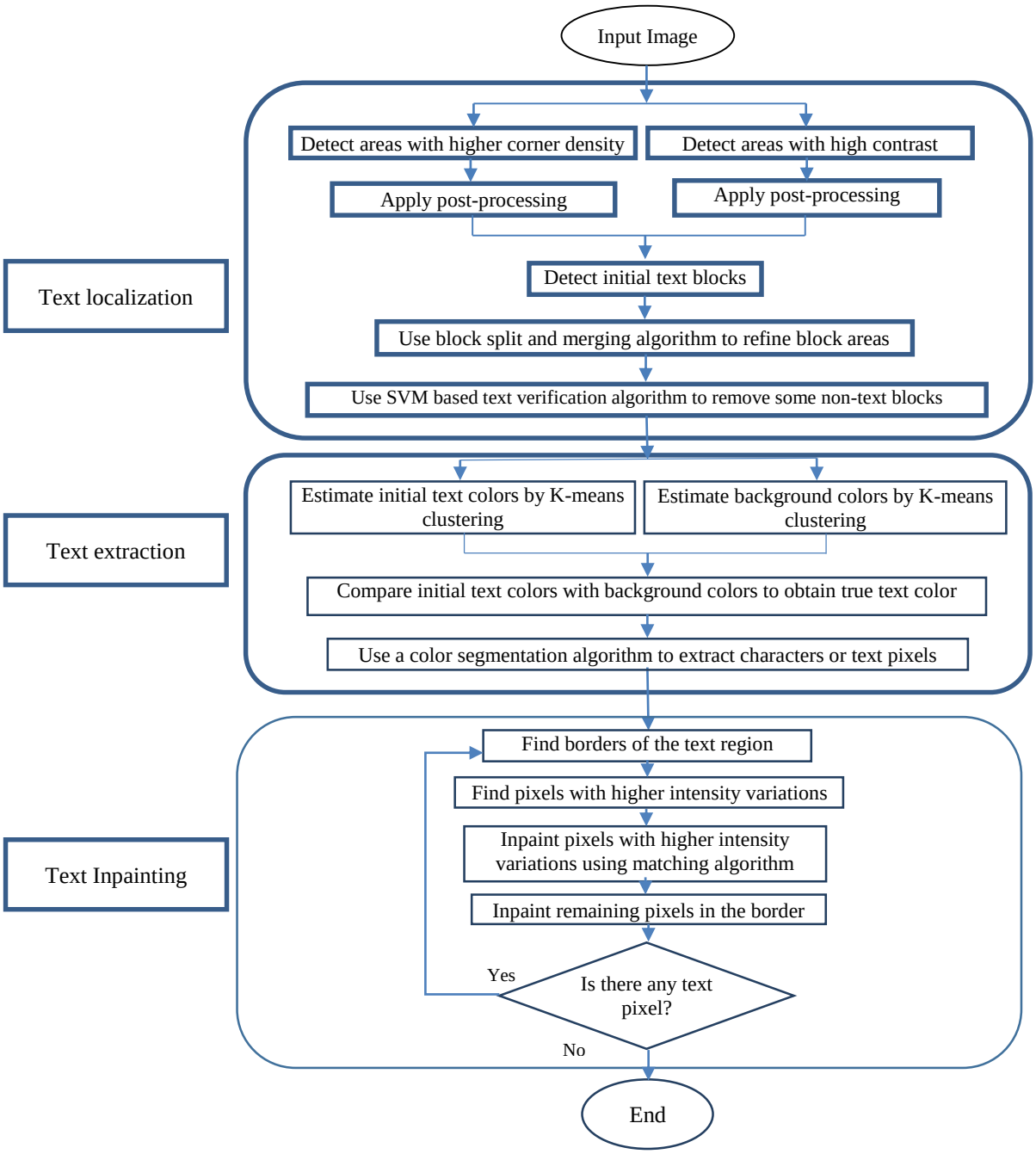

Fig. 1. General block scheme of the proposed algorithm.

A. *Localization of initial text blocks*

To localize initial text blocks, we detect areas with higher contrast and large number of corner points. For this purpose, we first apply a contrast enhancing operator to the image as follows:

$$Id(i,j) = \max\left(I(i,j) - \min(M), \max(M) - I(i,j)\right) \quad (1)$$

where M is a $m \times m$ window centered on image pixel (i,j). In our implementation, 9×9 windows are experimentally employed to calculate contrast enhanced image Id. Then, a threshold is applied to detect pixels with higher intensity contrast. We use an adaptive approach to calculate the contrast threshold value $cthr$ as follows:

$$I_{dt} = \begin{cases} Id & Id > 50 \\ 0 & o.w. \end{cases} \quad (2)$$

$$cthr = median\left(I_{dt}\right) \quad (3)$$

where the *median* function calculates the median value of nonzero pixels in I_{dt}. By applying the threshold to I_{dt}, the text location image using contrast enhancing operator i.e. I_{TL1} is obtained as:

$$I_{TL1} = \begin{cases} 1 & I_{dt} > cthr \\ 0 & o.w. \end{cases} \quad (4)$$

We also detect small elements in I_{TL1} and remove them as noisy pixels. Text locations also include corners with higher density. To employ this property for enhancing the efficiency of the text localization algorithm, image corners are extracted using Harris algorithm [55] and binary corner image is constructed as:

$$I_{corner}(i,j)=\begin{cases}1 & if\ (i,j)\ is\ a\ corner\\ 0 & o.w.\end{cases} \quad (5)$$

We then use morphological dilation operator to extend corner points and detect areas with higher corner density as follows:

$$I_{TL2} = I_{corner} \oplus B \quad (6)$$

where B is the structuring element and I_{TL2} is the text location image using corner densities. By combining the information of text location images, the initial text location image i.e. I_{TL} is calculated as follows:

$$I_{TL} = I_{TL1} \& I_{TL2} \quad (7)$$

where & represents logical-and operation. Fig. 2 shows a typical image and the resultant initial text location image.

In the next step of the algorithm, we divide I_{TL} into text blocks. To this end, we scan I_{TL} with 100×200 windows with the overlap of 50×100 pixels. Then the horizontal and vertical projections of binary pixels are used to divide the blocks into smaller blocks. We detect local minimums in the projection profiles to divide blocks. We then keep sub-blocks with enough number of pixels and repeat the split process again to obtain proper blocks. Fig. 3 shows four stages of the splitting process for extracting text blocks.

After extracting text blocks using the block splitting algorithm, we calculate the density and the area of the blocks to keep proper text blocks. For this purpose, we keep only blocks with the following conditions:

- Blocks with the density of more than 0.75.
- Blocks with the length and height of more than 8 pixels.
- Blocks with the area of more than 200 pixels.

We then merge blocks with overlap or very close blocks. In the case of merging, the bounding box of merged blocks is considered as a new block. Merging the blocks may create inappropriate blocks again; therefore we use horizontal and vertical projection profiles and repeat the splitting process again. We iterate splitting and merging process several times to extract proper text blocks. Fig. 4 shows the results of selecting initial text blocks after applying the split and merge algorithm.

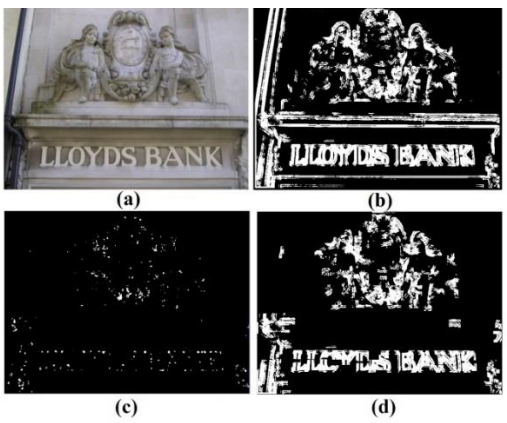

Fig. 2. Initial text location image. (a) Input image, (b) text pixels using contrast enhancing operator, (c) image corners using Harris operator, (d) ITL image.

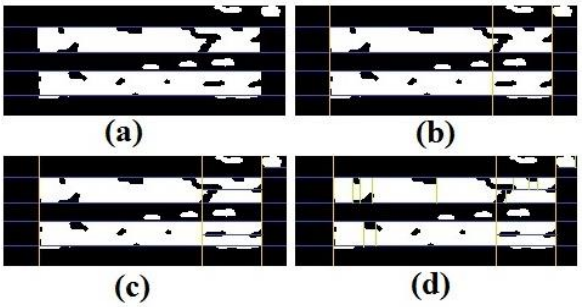

Fig. 3. Dividing ITL image into text blocks. (a) Dividing using horizontal projection profile, (b) dividing using vertical projection profile, (c) the second stage of splitting process using horizontal projection profile, (d) the second stage of splitting using vertical projection profile.

B. *Text verification*

The initial text blocks that are obtained in the first stage of text localization algorithm may also contain non-text areas. Therefore, we utilize verification stage to find and discard non-text areas. To verify text areas, a three-stage algorithm is utilized. In the first stage, we use the vertical and horizontal projection profiles of binary points in each candidate block at I_{TL}. We then apply a threshold to projection profiles using the mean and standard deviation of profiles as follows:

$$T_{ph} = m_{ph} + 0.5 * \sigma_{ph} \quad (8)$$

$$T_{pv} = m_{pv} + 0.5 * \sigma_{pv} \quad (9)$$

where m_{ph} and m_{pv} are the mean values of horizontal and vertical profile, σ_{ph} and σ_{pv} are the standard deviations and T_{ph} and T_{pv} are the threshold values for the horizontal and vertical profiles in each text blocks respectively. Then the number of profile elements which their value is higher than the threshold value is calculated. A block is considered as a text block if the following conditions are satisfied.

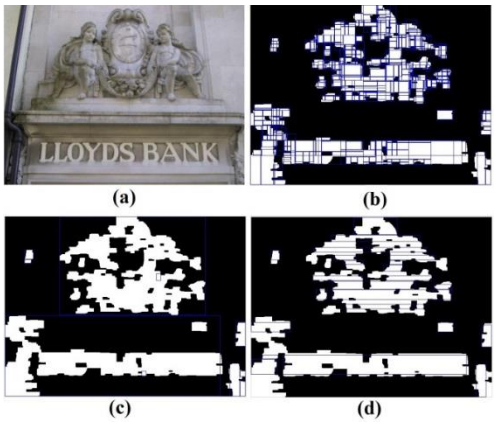

Fig. 4. The results of selecting initial text blocks. (a) Input image, (b) initial text blocks in ITL image after applying density and size constraints, (c) merged blocks, (d) splitting the blocks using projection profiles.

$$N_{ph} > 0.2h \quad (10)$$

$$N_{pv} > 0.4w \quad (11)$$

$$N_{pv} < 0.95w \quad (12)$$

where *h* and *w* are the block width and height respectively and N_{ph} and N_{pv} are the number of elements in horizontal and vertical profiles, which their values are greater than the predefined threshold values.

In the second stage of verification algorithm, we check the text block for the enough number of edge points. We extract edge points using Canny edge detector and discard blocks with few number of edge points.

The last stage of the text verification algorithm utilizes a SVM classifier with new set of textural features to remove non-text blocks completely. We use three sets of textural features to verify text blocks using SVM classifier as follows:

- Symmetry of text directions.
- Features based on vertical projection of the text pixels.
- Features based on co-occurrence matrix.

To calculate the first and second group features, it is necessary to extract text pixels or characters precisely. The algorithm to extract characters using the proposed color segmentation algorithm will be discussed in the next section. In other word the third stage of verification algorithm is employed after applying character extraction algorithm.

1) Symmetry of text directions

Edge directions in the outer boundaries of the characters are symmetric as shown in Fig. 5. We define the symmetry ratio of text pixels in the direction of θ as:

$$SR_\theta = \frac{\min(N_\theta, N_{\theta+180})}{\max(N_\theta, N_{\theta+180})} \quad (13)$$

where N_θ and $N_{\theta+180}$ are the number of pixels with the direction of θ and $\theta+180$ respectively.

To calculate the number of pixels in a given direction θ, the binary image representing text pixels is extracted using the color segmentation algorithm, which is described in the next section. Then by defining a proper mask and correlating the block image with the mask, the number of text pixels in the specified direction is calculated as follows:

$$I_\theta = I_B * W_\theta \quad (14)$$

$$I_{b\theta} = \begin{cases} 1 & \text{if } I_\theta \geq 5 \\ 0 & \text{o.w.} \end{cases} \quad (15)$$

$$N_\theta = \sum_j \sum_i I_{b\theta}(i, j) \quad (16)$$

where I_B is the binary block image after the extraction of text pixels and W_θ is the mask for detecting pixels in θ direction. We use 3×3 masks and $\theta = 0^\circ, -90^\circ$ for the feature extraction. Fig. 6 shows the required masks to detect pixels in $-90^\circ, 0^\circ, 90^\circ$, and 180° directions.

2) Features based on vertical projection of the text pixels

Fig. 7 shows the vertical projection profile of a sample Persian text. The profile includes several peaks and valleys, which are related to characters and spaces between them. We use this property as a feature to distinguish between text and non-text blocks.

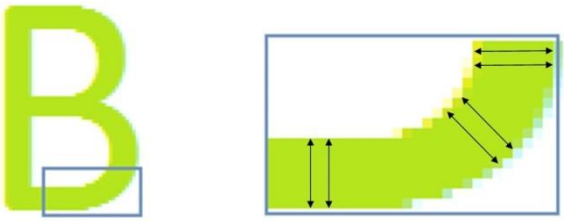

Fig. 5. Symmetry of pixel directions for B character.

$$0^\circ: \begin{pmatrix} -1 & -1 & -1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad 180^\circ: \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ -1 & -1 & -1 \end{pmatrix} \quad 90^\circ: \begin{pmatrix} 1 & 1 & -1 \\ 1 & 1 & -1 \\ 1 & 1 & -1 \end{pmatrix} \quad -90^\circ: \begin{pmatrix} -1 & 1 & 1 \\ -1 & 1 & 1 \\ -1 & 1 & 1 \end{pmatrix}$$

Fig. 6. 3×3 masks to detect pixels in $0^\circ, 90^\circ, -90^\circ$ and 180° directions.

To extract the feature, we first detect local maximums in the profile and apply a threshold to discard small peaks. The threshold value is set to 0.1 of the maximum peak value in the profile. We also remove very close peaks. Fig. 7-c shows the results of peak detection and thresholding algorithm. We extract some features like skewness, kurtosis and third order profile momentum for text verification.

3) Features based on co-occurrence matrix

To extract features based on co-occurrence matrix, we first calculate co-occurrence matrix for gray level pixel values in the text block. Co-occurrence matrix C is defined over an *m*×*n* text block image *I*, parameterized by an offset *(Δx, Δy),* as:

$$C_{\Delta x,\Delta y}(i, j) = \sum_{p=1}^{m} \sum_{q=1}^{n} \begin{cases} 1, & \text{if } I(p, q) = i \text{ and } I(p+\Delta x, q+\Delta y) = j \\ 0, & o.w. \end{cases} \quad (17)$$

We experimentally use *Δx=0* and *Δy=2* to construct co-occurrence matrix *C*. We then extract the following features for the text verification.

$$Entropy = \sum_i \sum_j C(i, j) \times \log(C(i, j)) \quad (18)$$

$$Contrast = \sum_i \sum_j (i-j)^2 C(i, j) \quad (19)$$

$$Energy = \sum_i \sum_j C^2(i, j) \quad (20)$$

$$Homogeneity = \sum_i \sum_j \frac{C(i, j)}{1+|i-j|} \quad (21)$$

Fig. 8 shows the results of the proposed text verification algorithm. As shown in the figure, the

proposed algorithm can successfully discard non-text blocks.

IV. TEXT EXTRACTION

The aim of the text extraction algorithm is to extract text pixels from the text blocks. It is assumed that all the text pixels in a text block have the same color; however the background may be complex. The aim of the proposed algorithm in this paper is to inpaint the extracted text pixels; therefore the precise extraction of the text pixels is mandatory for our algorithm. To this intent, we estimate text color and use the color segmentation algorithm to extract characters or text pixels.

Since the text block includes both text (foreground) and background pixels, we first estimate background colors and compare colors in the block with the background colors to estimate the true text color.

To estimate colors for background area, we consider two areas with the height of two pixels above and under the candidate text block as shown in Fig. 9-(b). Then k-means clustering algorithm is employed to cluster background colors. For clustering background color, we start by *K* initial clusters, which are estimated using the histogram of different color channels. After applying k-means clustering algorithm, several clusters with higher number of members are selected as the candidate background colors.

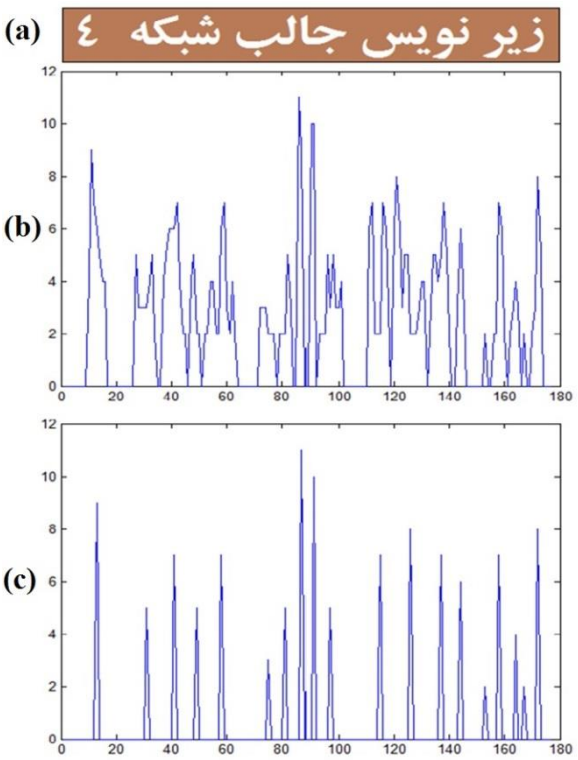

Fig. 7. Feature extraction using vertical projection profile of text pixels. (a) A text block, (b) vertical projection profile, (c) profile after peak detection and thresholding.

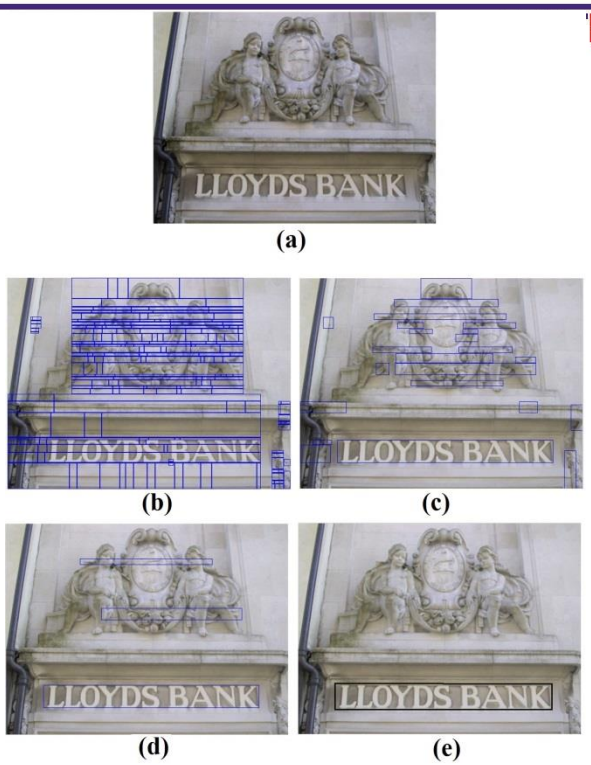

Fig. 8. Results of the proposed text verification algorithm. (a) Input image, (b) initial candidate text blocks, (c) text blocks after text verification using projection profile, (d) text verification using edge density constraint, (e) final text blocks after verification using the SVM classifier.

To estimate initial clusters for k-means algorithm, we calculate histogram of pixels for each color channel and after applying one-dimensional smoothing filter and thresholding algorithm; the local maximums in the histogram are employed to determine initial clusters.

We also employ k-means clustering algorithm to partition different colors in the text block into several clusters. These clusters include both text and background colors. Therefore by comparing the color clusters of background pixels with color clusters of the text block, the similar colors are removed and the cluster of the text block which has the maximum Euclidian distance from the clusters of the background is selected as the text color.

To increase the efficiency of the proposed algorithm for estimating text color, we also divide the text block into several sub-blocks and estimate text color in each sub-blocks. Then, the clusters with similar colors are merged and finally the cluster with maximum number of members is selected as the text color.

When the text color estimated, a color segmentation based on the Euclidian distance is employed to extract text pixels or characters. In this method a pixel is considered as text pixel, if the Euclidian distance between its color and the estimated text color is less than the color radius *R*. *R* value has the major impact on the accuracy of the text extraction algorithm. To determine *R* value, the minimum distance between the selected text color cluster and clusters related to background colors is calculated. We then estimate *R* value based on this distance.

Fig. 9 shows a typical text block, clusters for background, the text block and the extracted text pixels. As shown in the figure, the algorithm can extract text pixels successfully even in low contrast images.

V. TEXT INPAINTING

The proposed inpainting algorithm is based on texture synthesis and matching and the priority of the damaged pixels. The algorithm finds the best matching pixels for the damaged pixels and repairs them based on the matched pixels. The proposed inpainting algorithm includes the following stages:

- Find the pixels in the border of the damaged region by using morphological erosion operator.
- To retain the structure of the image in the damaged region, find pixels with high intensity variation and inpaint using the matching algorithm.
- Inpaint the remaining pixels in the borders using the matching algorithm and priority based on less damaged pixels in the 8-neighbors.
- Repeat the inpainting algorithm to fully reconstruct the damaged region.

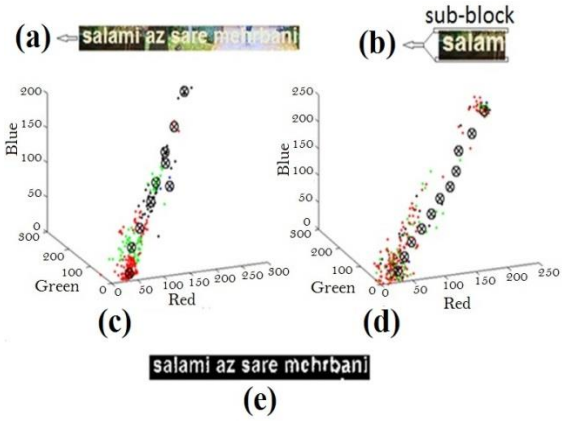

Fig. 9. Results of text extraction algorithm. (a) A typical text block, (b) background area, (c) the result of clustering text color, (d) the result of clustering background color, (e) the result of color segmentation.

To inpaint a damaged pixel, the matching algorithm searches in 8×8 windows centered on the selected damaged pixel for the best matching pixel. The best matching pixel is a pixel, which is more similar to the damaged pixel. To measure similarity, we use Sum Squared Difference (SSD) criterion as follows:

$$r(d_x,d_y)= \sum_{c=R,G,B}\sum_{x=-w/2}^{x=+w/2}\sum_{y=-w/2}^{y=+w/2}(I_c(x,y)-I_c(x+d_x,y+d_y))^2 \quad (22)$$

where $r(d_x,d_y)$ represent the similarity value, w is the window size to measure similarity and c is the index of the color channel.

To detect pixels with high variations, we calculate the intensity variation using 3×3 windows and SSD approach. Then, we apply a threshold to determine pixels with high variations. The threshold value is calculated based on the maximum value of intensity variations.

Fig. 10 shows several iterations of the inpainting algorithm using the texture matching algorithm with the proposed priority scheme and without it. As shown in the figure, the proposed priority scheme retains image structures efficiently.

VI. EXPERIMENTAL RESULTS

The proposed algorithm was implemented using a MATLAB program and tested using several images and video frames. To test the implemented algorithm, we utilized a Personnel Computer (PC) with Microsoft Windows 7 OS and 2.8GHZ Core Duo 2 CPU and 6MB cache.

To test the proposed algorithms, we provided a database of image and video frames with Arabic, English and Korean subtitles from the internet.

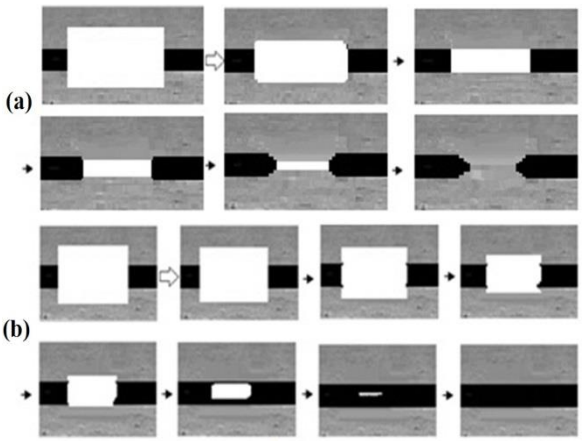

Fig. 10. The results of inpainting algorithm using texture matching. (a) Standard texture matching, (b) the proposed algorithm with the priority scheme.

Table I shows the results of text localization algorithm. In this table, the results of the proposed algorithm are shown for the collected database with different text languages including English, Arabic and Korean. In this table, the results of text localization algorithm based on Laplacian [18], corner [24] , Discrete Cosine Transform (DCT) [20] and end-to-end text recognition [27] are also shown for the comparison. In this table, NI, DB, FP, TP, FN, DR, FAR and FRR stand for the Number of Images, Detected Blocks, False Positive, True Positive, False Negative, Detection Rate, False Acceptance Rate and False Rejection Rate respectively. DR, FAR and FRR are defined as follows [56]:

$$DR = \frac{TP}{TP+FN} \quad (23)$$

$$FAR = \frac{FP}{TP+FN} \quad (24)$$

$$FRR = \frac{FN}{TP+FN} \quad (25)$$

As Table I shows, the proposed algorithm is more efficient in detecting text blocks. The results of Table I shows that only the results of end-to-end text recognition algorithm [27] is comparable with the results of our method. However its computation burden is very high in comparison with our method. The

average processing time for end-to-end text recognition algorithm is about 25 minutes for each image. However, the proposed method can process each image in 20 seconds.

Fig. 11 shows the results of the text extraction algorithm. The figure shows some candidate text blocks and the extracted characters. The results of Niblack [31], Sauvola [33] and Otsu [34] methods are also shown in this figure for comparison. As the figure shows the proposed algorithm outperforms the Sauvola, Niblack and Otsu methods.

Table II illustrates the results of various text extraction algorithms numerically. The table shows the efficiency of the proposed algorithm.

Fig. 12 shows the results of the proposed inpainting algorithm on several frames of a video file. As it is shown in the figure, the reconstructed area is not recognizable visually. Fig. 13 compares the results of the proposed inpainting algorithm with the Total Variational (TV) algorithm [38] with 1000 iterations and method based on coherence transport [47]. The figure shows less blurring effect for the proposed algorithm.

TABLE I. THE RESULTS OF DIFFERENT TEXT LOCALIZATION ALGORITHMS.

Method	Language	NI	DB	FP	TP	FN	FAR	FRR	DR
Localization based on Laplacian [18]	English	40	76	30	46	24	0.408	0.316	0.605
	Korean	50	88	10	78	10	0.114	0.114	0.886
	Arabic	50	62	0	62	4	0	0.064	1
Localization based on corner [24]	English	40	77	15	62	4	0.195	0.052	0.880
	Korean	50	86	10	76	3	0.116	0.035	0.884
	Arabic	50	62	0	62	0	0	0	1
Localization based on DCT [20]	English	40	74	30	44	7	0.588	0.137	0.595
	Korean	50	291	167	124	51	0.954	0.291	0.426
	Arabic	50	195	70	125	16	0.496	0.113	0.641
End-to-end text recognition [27]	English	40	64	1	63	0	0.016	0	1
	Korean	50	81	1	80	2	0.012	0.025	0.976
	Arabic	50	65	4	61	0	0.065	0	1
Proposed method	English	40	66	1	65	0	0.015	0	1
	Korean	50	79	0	79	1	0	0.012	0.988
	Arabic	50	62	0	62	1	0	0.016	0.984

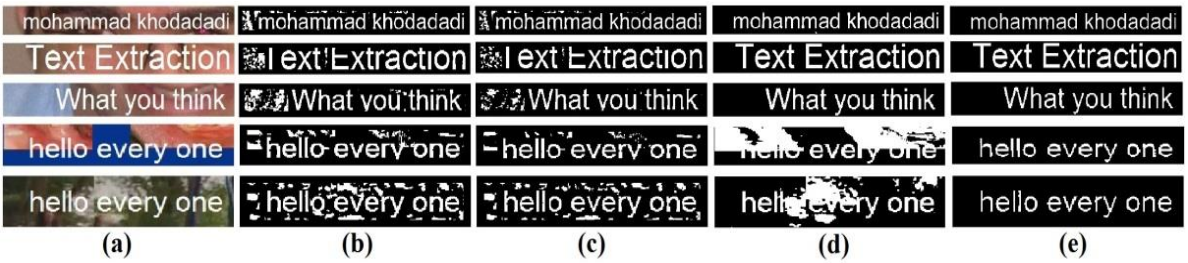

Fig. 11. The results of text extraction algorithm. (a) Candidate text block; extracted characters by (b) Niblack [31] , (c) Sauvola [33], (d) Otsu [34] and (e) the proposed method.

To measure the distortion generated by the proposed inpainting algorithm, we collected a database of images by adding some text blocks to 100 video frames. Then after applying the proposed text localization, extraction and inpainting algorithm, the difference between the reconstructed and original image is employed to measure the distortion of the inpainting algorithm. To measure the efficiency of different inpainting algorithms, we calculate Peak Signal-to-Noise Ratio (PSNR), Mean Squared Error (MSE), Minimum Error (MinError) and Median of Errors (MedError) as follows [57]:

$$MSE = \frac{1}{3\times m\times n}\sqrt{\sum_{c=R,G,B}\sum_{j=0}^{n-1}\sum_{i=0}^{m-1}(I^{c}_{org}(i,j)-I^{c}_{rcnst}(i,j))^{2}} \quad (26)$$

$$MinError = \min_{c,i,j}\left(\left|I^{c}_{org}(i,j) - I^{c}_{rcnst}(i,j)\right|\right) \quad (27)$$

$$MedError = \underset{c,i,j}{median}\left(\left|I^{c}_{org}(i,j) - I^{c}_{rcnst}(i,j)\right|\right) \quad (28)$$

$$PSNR = 20\log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \quad (29)$$

Here m and n are image width and height respectively, I^{c}_{org} and I^{c}_{rcnst} are the original and the reconstructed images for color channel c respectively and MAX_I is the maximum possible pixel value of the image.

TABLE II. THE RESULTS OF DIFFERENT TEXT EXTRACTION ALGORITHMS.

Character extraction method	NIB	FAR	FRR	DR	Processing time (sec)
Niblack [31]	100	100.18%	4.30%	95.70%	0.60
Sauvola [33]	100	74.32%	4.54%	95.46%	0.58
Otsu [34]	100	103.35%	2.40%	97.6%	0.15
Proposed method	100	4.88%	1.97%	98.03%	2.65

Fig. 12. The results of text localization, extraction and inpainting algorithms (a) original images, (b) the extracted texts, (c) the reconstructed images after applying the proposed text extraction and inpainting algorithm.

Fig. 13. The results of different inpainting algorithms. (a) Damaged image, (b) the results of TV method [38] with 1000 iterations, (c) inpainting based on coherence transport [47], (d) the proposed inpainting algorithm.

TABLE III. THE RESULTS OF DIFFERENT INPAINTING ALGORITHMS.

Inpainting Method	Processsing time (sec)	Quality measurement metric			
		MSE	MinError	MedError	PSNR
TV method [38]	30.4	0.015	0.000006	0.02	18.15
Coherence transport [47]	3.2	0.0129	0	0.01	19.89
Proposed method	18.8	0.0082	0	0.02	20.87

Fig. 14. Source of errors for the proposed text localization algorithm. (a) Error of text verification stage, (b) error because of low contrast texts and holes in texts and (c) error because of separating nearby text areas using block split and merging algorithm.

Table III compares the efficiency of different text inpainting algorithms using different image quality measurement metrics. The table indicates a considerable enhancement for the proposed inpainting algorithm.

We analyzed the source of error for the proposed text localization and text extraction algorithms. Fig. 14 shows three images with erroneous text blocks localization. Briefly three main sources of error for the proposed text localization and text extraction algorithms are as follows:

- The error of text verification algorithm. This error is generally originated by SVM based text verification algorithm. Fig.14 (a) shows a typical errors text verification algorithm. As shown in this figure, some non-text areas are not discarded by text verification algorithm.
- Errors because of low contrast texts and holes in characters. As shown in Fig. 14 (a) and (b) some text areas may not have enough contrast or there may some holes in characters (Fig. 14 (b)). This results in non-detection of some text areas or errors in separating text blocks.
- Error because of separating nearby text areas using block split and merging algorithm. Figure 14 (c) shows a sample error of block split and merging algorithm. This error is generally occurs in Korean and Arabic texts. This error is generally originated because of non-uniform text intervals in horizontal and vertical directions.

VII. Conclusions

In this paper, new methods for text localization, extraction and inpainting are proposed. The text localization algorithm is based on image gradient and new sets of textural features. The text localization algorithm employs different stages of text verification to remove non-text blocks. To extract characters precisely, we estimate text and background colors and use a color segmentation algorithm to extract characters. Finally a text inpainting algorithm is proposed to reconstruct the original image. The inpainting algorithm is based on texture synthesis and priority based on image structure. We tested the proposed algorithms with different images and compared the results with those of other methods. The results showed the efficiency of the proposed algorithms for text localization, extraction and inpainting. As a future work, we plan to develop an algorithm to use temporal information of video frames for text localization, extraction an inpainting.

References

[1] W. Fan, J. Sun, Y. Katsuyama, Y. Hotta, and S. Naoi, "Text Detection in Images Based on Grayscale Decomposition and Stroke Extraction," in *2009 Chinese Conference on Pattern Recognition (CCPR 2009)* 2009, pp. 1-4.

[2] D. Q. Zhang and S. F. Chang, "Learning to detect scene text using a higher-order MRF with belief propagation," in *Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04)* 2004, pp. 101-108.

[3] J. D. Kim, Y. J. Han, and H. Hahn, "License Plate Detection Using Topology of Characters and Outer Contour," in *Second International Conference on Future Generation Communication and Networking Symposia (FGCNS'08)* 2008, pp. 171-174.

[4] J. Zhang and R. Kasturi, "Text detection using edge gradient and graph spectrum," in *20th International Conference on Pattern Recognition (ICPR)*, 2010, pp. 3979-3982.

[5] K. L. Bouman, G. Abdollahian, M. Boutin, and E. J. Delp, "A Low Complexity Sign Detection and Text Localization Method for Mobile Applications," *IEEE Transactions on Multimedia*, vol. 13, pp. 922-934, 2011.

[6] J. Gllavata, E. Qeli, and B. Freisleben, "Detecting text in videos using fuzzy clustering ensembles," in *Eighth IEEE International Symposium on Multimedia (ISM'06)* 2006, pp. 283-290.

[7] Y. P. Huang, L. W. Hsu, and F. E. Sandnes, "An intelligent subtitle detection model for locating television commercials," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, pp. 485-492, 2007.

[8] M. Khodadadi and A. Behrad, "Text Localization, Extraction and Inpainting in Color Images," in *20th Iranian Conference on Electrical Engineering (ICEE)* 2012, pp. 1035-1040

[9] W. Kim and C. Kim, "A new approach for overlay text detection and extraction from complex video scene," *IEEE Transactions on Image Processing*, vol. 18, pp. 401-411, 2009.

[10] X. Li, W. Wang, Q. Huang, W. Gao, and L. Qing, "A hybrid text segmentation approach," in *IEEE International Conference on Multimedia and Expo (ICME 2009)*, 2009, pp. 510-513.

[11] R. Lienhart and A. Wernicke, "Localizing and segmenting text in images and videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, pp. 256-268, 2002.

[12] Q. Liu, C. Jung, S. Kim, Y. Moon, and J. Kim, "Stroke filter for text localization in video images," in *IEEE International Conference on Image Processing (ICIP2006)* 2006, pp. 1473-1476.

[13] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H. 264/AVC video compression standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 620-636, 2003.

[14] G. Miao, Q. Huang, S. Jiang, and W. Gao, "Coarse-to-fine video text detection," in *IEEE International Conference on Multimedia and Expo (ICME2008)*, 2008, pp. 569-572.

[15] Y. F. Pan, X. Hou, and C. L. Liu, "A hybrid approach to detect and localize texts in natural scene images," *IEEE Transactions on Image Processing*, vol. 20, pp. 800-813, 2011.

[16] L. Sang and J. Yan, "Rolling and non-rolling subtitle detection with temporal and spatial analysis for news video," in *2011 International Conference on Modelling, Identification and Control (ICMIC)*, 2011, pp. 285-288.

[17] P. Shivakumara, T. Q. Phan, and C. L. Tan, "New Fourier-statistical features in RGB space for video text detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, pp. 1520-1532, 2010.

[18] P. Shivakumara, T. Q. Phan, and C. L. Tan, "A Laplacian approach to multi-oriented text detection in video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 412-419, 2011.

[19] X. Wang, "The Research of Subtitles Regional Location Algorithm Based on Video Caption Frames," in *Second International Symposium on Intelligent Information Technology Application (IITA'08)* 2008, pp. 886-889.

[20] Z. Xu, H. Ling, F. Zou, Z. Lu, P. Li, and T. Wang, "Fast and robust video copy detection scheme using full DCT coefficients," in *IEEE International Conference on Multimedia and Expo (ICME 2009)*, 2009, pp. 434-437.

[21] Q. Yang, "An improved algorithm of news video caption detection and recognition," in *2011 International Conference on Computer Science and Network Technology (ICCSNT)*, , 2011, pp. 1549-1552.

[22] C. Yi and Y. L. Tian, "Text string detection from natural scenes by structure-based partition and grouping," *IEEE Transactions on Image Processing*, vol. 20, pp. 2594-2605, 2011.

[23] B. Zafarifar and J. Cao, "Instantaneously responsive subtitle localization and classification for TV applications," *IEEE Transactions on Consumer Electronics*, vol. 57, pp. 274-282, 2011.

[24] X. Zhao, K. H. Lin, Y. Fu, Y. Hu, Y. Liu, and T. S. Huang, "Text from corners: a novel approach to detect text and caption in videos," *IEEE Transactions on Image Processing*, vol. 20, pp. 790-799, 2011.

[25] L. Chunmei, "Text extraction from natural images based on stroke width map," in *2013 IEEE Second International Conference on Image Information Processing (ICIIP)*, Shimla, India, 2013, pp. 556-559.

[26] B. Shekar, M. Smitha, and P. Shivakumara, "Discrete Wavelet Transform and Gradient Difference based approach for text localization in videos," in *2014 Fifth International Conference on Signal and Image Processing (ICSIP)*,, Bangalore, India, 2014, pp. 280-284.

[27] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *2012 21st International Conference on Pattern Recognition (ICPR)*, 2012, pp. 3304-3308.

[28] Y. Xu-Cheng, Y. Xuwang, H. Kaizhu, and H. Hong-Wei, "Robust Text Detection in Natural Scene Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, pp. 970-983, 2014.

[29] P. Shivakumara, A. Dutta, C. L. Tan, and U. Pal, "Multi-oriented scene text detection in video based on wavelet and angle projection boundary growing," *Multimedia tools and applications*, vol. 72, pp. 515-539, 2014.

[30] X.-C. Yin, W.-Y. Pei, J. Zhang, and H.-W. Hao, "Multi-Orientation Scene Text Detection with Adaptive Clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, pp. 1-1, 2015.

[31] W. Niblack, *An introduction to digital image processing*, 2 ed.: Prentice-Hall, 1985.

[32] C. Wolf, J. M. Jolion, and F. Chassaing, "Text localization, enhancement and binarization in multimedia documents," in *16th International Conference on Pattern Recognition (ICPR2002)*, 2002, pp. 1037-1040.

[33] J. Sauvola and M. Pietikäinen, "Adaptive document image binarization," *Pattern Recognition*, vol. 33, pp. 225-236, 2000.

[34] Y. Hao and F. Zhu, "Fast Algorithm for Two-dimensional Otsu Adaptive Threshold Algorithm [J]," *Journal of Image and Graphics*, vol. 4, p. 014, 2005.

[35] B. F. Wu, S. P. Lin, and C. C. Chiu, "Extracting characters from real vehicle licence plates out-of-doors," *IET Computer Vision*, vol. 1, pp. 2-10, 2007.

[36] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, "Filling-in by joint interpolation of vector fields and gray levels," *IEEE Transactions on Image Processing*, vol. 10, pp. 1200-1211, 2001.

[37] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *27th annual conference on Computer graphics and interactive techniques*, 2000, pp. 417-424.

[38] J. Shen and T. F. Chan, "Mathematical models for local nontexture inpaintings," *SIAM Journal on Applied Mathematics*, vol. 62, pp. 1019-1043, 2002.

[39] M. Bertalmio, "Strong-continuation, contrast-invariant inpainting with a third-order optimal PDE," *IEEE Transactions on Image Processing*, vol. 15, pp. 1934-1938, 2006.

[40] J. M. Fadili and G. Peyré, "Total variation projection with first order schemes," *IEEE Transactions on Image Processing*, vol. 20, pp. 657-669, 2011.

[41] T. F. Chan and J. Shen, "Nontexture inpainting by curvature-driven diffusions," *Journal of Visual Communication and Image Representation*, vol. 12, pp. 436-449, 2001.

[42] P. Elango and K. Murugesan, "Digital Image Inpainting Using Cellular Neural Network," *Int. J. Open Problems Compt. Math*, vol. 2, pp. 439-450, 2009.

[43] M. N. Favorskaya, A. G. Zotin, and M. V. Damov, "Intelligent inpainting system for texture reconstruction in videos with text removal," in *2010 International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 2010, pp. 867-874.

[44] H. Guo, N. Ono, and S. Sagayama, "A structure-synthesis image inpainting algorithm based on morphological erosion operation," in *2008 Congress on Image and Signal Processing (CISP'08)* 2008, pp. 530-535.

[45] D. Liu, X. Sun, F. Wu, S. Li, and Y. Q. Zhang, "Image compression with edge-based inpainting," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, pp. 1273-1287, 2007.

[46] J. Park, D. C. Park, R. J. Marks, and M. A. El-Sharkawi, "Recovery of image blocks using the method of alternating projections," *IEEE Transactions on Image Processing*, vol. 14, pp. 461-474, 2005.

[47] F. Bornemann and T. März, "Fast image inpainting based on coherence transport," *Journal of Mathematical Imaging and Vision*, vol. 28, pp. 259-278, 2007.

[48] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Transactions on Image Processing*, vol. 13, pp. 1200-1212, 2004.

[49] L. Demanet, B. Song, and T. Chan, "Image inpainting by correspondence maps: a deterministic approach," *Applied and Computational Mathematics*, vol. 1100, pp. 217-50, 2003.

[50] E. A. Pnevmatikakis and P. Maragos, "An inpainting system for automatic image structure-texture restoration with text removal," in *15th IEEE International Conference on Image Processing (ICIP 2008)*, 2008, pp. 2616-2619.

[51] T. H. Tsai and C. L. Fang, "Text-Video Completion Using Structure Repair and Texture Propagation," *IEEE Transactions on Multimedia*, vol. 13, pp. 29-39, 2011.

[52] Y. Wu, M. Wang, and X. Liu, "A Fast Inpainting Algorithm Using Half-Point Gradient," in *Second International Workshop*

on Computer Science and Engineering (WCSE'09) 2009, pp. 594-598.

[53] Z. Xu and J. Sun, "Image inpainting by patch propagation using patch sparsity," *IEEE Transactions on Image Processing*, vol. 19, pp. 1153-1165, 2010.

[54] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting," *IEEE Transactions on Image Processing*, vol. 12, pp. 882-889, 2003.

[55] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey vision conference*, 1988, p. 50.

[56] I. Chingovska, A. Anjos, and S. Marcel, "Anti-spoofing: Evaluation Methodologies," in *Encyclopedia of Biometrics*, ed: Springer, 2014, pp. 1-6.

[57] K.-H. Thung and P. Raveendran, "A survey of image quality measures," in *2009 International Conference for Technical Postgraduates (TECHPOS)*, 2009, pp. 1-4.

Mohammad Khodadadi Zadboni received the B.Sc. degree in electronic engineering from Mazandaran University of Technology, Mazandaran, Iran, in 2004, and the M.Sc. degree in electronic engineering (image processing) from Shahed University, Tehran, Iran, in 2009. He is now a researcher in ITRC, Tehran, Iran. His research interests include image processing, computer vision, biomedical image processing and pattern recognition.

Alireza Behard was born in Iran in 1973. He received the B.Sc. degree in electronic engineering from Electrical Engineering Faculty, Tabriz University, Tabriz, Iran, in 1995. In 1998, he received his M.Sc. degree in digital electronics from Electrical Engineering Faculty, Sharif University of Technology, Tehran, Iran. He received his Ph.D. degree in electronic engineering from Electrical Engineering Faculty, Amirkabir University of Technology, Tehran, Iran, in 2004. Currently, he is an associate professor of Electrical and Electronic Engineering Department, Shahed University, Tehran, Iran. His research fields are image and video processing and machine vision with special attention to visual target tracking.

This Page intentionally left blank.