IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 9, 2020, accepted October 1, 2020, date of publication October 6, 2020, date of current version October 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3028903

Directional Coherence-Based Scrolling-Text Detection for Frame Rate Up-Conversion

HO SUB LEE[1], (Member, IEEE), AND SUNG IN CHO[2], (Member, IEEE)
[1]Department of Electrical Engineering, Kyungnam University, Changwon 51767, South Korea
[2]Department of Multimedia Engineering, Dongguk University, Seoul 04620, South Korea

Corresponding author: Sung In Cho (csi2267@dongguk.edu)

ABSTRACT This article proposes a new scrolling-text detection method that uses directional coherence for frame rate up-conversion (FRUC). Most previous methods use either gradient information or motion vector (MV) distribution of the frame for scrolling-text detection. Edges can be generated by non-text components and the number of MVs to determine the scrolling-text decreases in each row of the frame. Thus, they incorrectly detect the non-text regions as scrolling-text and cannot accurately detect the start or end of text scrolling at the frame boundary. The proposed method overcomes these problems using coherence values of edge directions for each pixel and scrolling-text-aware refinement processes. The key idea of the proposed method is to use the directional coherence of edge directions and use texture patterns analysis-based refinement to improve the accuracy of the scrolling-text detection. For refinement processes, the proposed method extracts texture patterns as bit codes. Then, it computes the diversity of the texture patterns around the detected text edges. In addition, the proposed method extracts the representative value of the MV for the detected region to correct the regions falsely detected as the scrolling-text. With these refinement processes, the proposed method can also accurately detect the start or end of text scrolling at the frame boundary. In the experimental results, the proposed method increased the average F_1 score to 0.504 (a 131.25% improvement) compared with previous methods. The average computation time per pixel of the proposed method also decreased to 18.571 μs (an 80.80% reduction) compared with previous methods.

INDEX TERMS Frame rate up-conversion, coherence of edge directions, scrolling-text detection, motion estimation.

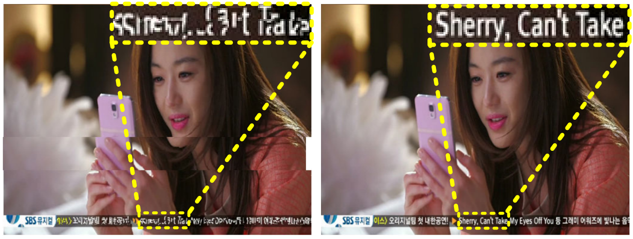
I. INTRODUCTION

Frame rate up-conversion (FRUC) is a technique that increases the frame rate of videos by inserting interpolated frames between two consecutive frames [1]–[4]. Interpolated frames are generated using motion vectors (MVs) between two consecutive frames. FRUC has been used for film-to-video conversion to increase the frame rate of films [5]. It is also used in liquid crystal displays (LCDs) [6] to reduce motion blur and TV standard conversion with different frame rates [7]. Because the fps of the input frame is varies, FRUC is an essential technique for display systems with a predetermined fps.

FRUC consists of two primary steps [1]–[4]: motion estimation (ME) and motion-compensated interpolation (MCI). ME calculates MVs of an object between two consecutive frames and corrects the outliers in a set of MVs. MCI produces a new interpolated frame between two original frames using the calculated MVs. ME is the most important of the FRUC steps because FRUC performance is highly dependent on the accuracy of the MVs calculated by the ME. FRUC can generate interpolated frames with artifacts in the regions where MV is not accurately measured. Many cases exist in which conventional FRUC cannot generate accurate MVs. In these cases, conventional FRUC often fails to extract accurate MVs in the scrolling-text regions. Artifacts in the scrolling-text regions are easier to recognize than artifacts in other regions (Fig. 1). This is because the scrolling-text provides vital information to the viewers. It is also easily recognized by the human eye compared with other regions. Therefore, the scrolling-text detection process is essential for FRUC to perform correction of the MVs in the scrolling-text regions to improve the quality of interpolated frames.

Generally, scrolling-text detection method consists of the following steps. In the first step, it generates a text map. Most previous methods use the edge magnitude of pixels

The associate editor coordinating the review of this manuscript and approving it for publication was Zhen Ren.



(a) (b)

FIGURE 1. Comparison of interpolated frames: (a) interpolated frame with artifacts in the scrolling-text region (FRUC without scrolling-text detection) and (b) interpolated frame without artifacts in the scrolling-text region (FRUC with the proposed method). ©Love from Star SBS.

and motion information in consecutive frames to quantify the scrolling-text position in the video contents. Existing edge detection methods [8]–[13] are widely used to extract the edge magnitude of pixels. In the second step, it refines the text map to determine the final scrolling-text regions.

In this article, we propose a directional coherence-based scrolling-text detection method for FRUC. It generates a text map using the calculated directional coherence values for each pixel in the frame. We exploit directional coherence values to estimate the degree of coherency of gradient orientations. Our method generates an initial scrolling-text map by calculating the difference between the current and previous text maps. Then, the proposed method refines the initial scrolling-text map by analyzing the text edge density and Local Directional Texture Pattern (LDTP) [14] of the detected initial scrolling-text map. The mixture of edge direction and texture patterns is used to determine the candidate regions as the scrolling-text regions. Furthermore, we use the MV distribution-based refinement method to correct the regions falsely detected as the scrolling-text regions. The three contributions of this article are as follows:

1) The proposed method uses a directional coherence concept to detect the text edges. The use of directional coherence can distinguish the text edges from highly-textured regions or uniformly-textured (flat) regions.
2) We use a scrolling-text-aware refinement process, which is based on the texture patterns analysis, to improve the accuracy of the scrolling-text detection. Based on the observation that the luminance variation around the text pixels is large, the proposed method can detect the scrolling-text regions using the diversity of the texture patterns.
3) We verified the performance of the proposed method by comparing the interpolated frames generated by the conventional FRUC algorithm [15].

The remainder of this article is organized as follows. Section II shows the brief review of previous methods for text detection. Section III describes the proposed scrolling-text detection method. In Section IV, we compared the scrolling-text detection accuracy of previous methods with that of the proposed method. Furthermore, Section IV presents a quality evaluation of the interpolated frames generated by conventional FRUC. Finally, Section V concludes the paper.

II. RELATED WORK

Text, which are found on natural scenes or video sequences, can be categorized into two types: static-text and scrolling-text. Various static-text detection methods in natural scenes or video sequences have been widely investigated [16]–[20]. To identify text regions, Delaunay triangulation-based text detection method [16] that uses symmetrical features was proposed. It detects the text edges using the fact that text edges have many parallel edges. Then, this method detects the candidate text regions by forming Delaunay triangulation for corners of the edge map. Ring Radius Transform (RRT)-based method [17] detects the multi-oriented text in natural scenes. Histogram Oriented Moment (HOM)-based method [18] extracts connected components and identifies text components using a Support Vector Machine (SVM) classifier. Then, Recurrent Neural Network (RNN)-based classifier is used for recognition of text. Fractals-based text detection method was proposed [19]. It uses fractal properties in the gradient domain and separates text components from non-text components. Fourier-Laplacian transform-based text detection method [20] that includes a verification technique using Hidden Markov Model (HMM) extracts the candidate text region. After extracting the text candidate regions, it verifies the final text regions using the HMM-based classification. The above existing methods are mainly related to static-text detection in natural scenes or video sequences. Unlike these methods, this article focuses on developing the scrolling-text detection method with emphasis on the FRUC application.

Scrolling-text detection is process of detecting the scrolling-text from a video to reduce the artifacts that are frequently caused by scrolling-text in FRUC. Several scrolling-text detection methods have been proposed [21]–[25]. The edge information-based scrolling-text detection method [21] detects scrolling-text depicted on the horizontal or vertical regions of the frame boundary by projecting the edge information obtained from a Sobel edge detector [12] on the horizontal or vertical axis. This method exploits the concept that scrolling-text regions have a high edge density. It detects scrolling-text accurately if the scrolling-text exists entirely in the horizontal or vertical direction on a simple background of the given frame. However, this method incorrectly detects highly-textured regions as scrolling-text because the edges generated by non-text components can be included in the projection results. Furthermore, it cannot accurately detect scrolling-text when it begins to scroll from the frame boundary or when it ends scrolling at the frame boundary because edge density described by the projection for these types of text is low.

The MV distribution-based scrolling-text detection method [22] extracts the MVs in each row of the frame and finds dominant MV directions for each row of the frame to detect scrolling-text. This method is straightforward because it only extracts the MV direction in each row of the frame boundary and can be applied easily to FRUC. However, because the number of MVs available to determine the scrolling-text regions decreases in each row of the frame boundary, it is difficult to extract the MV direction of scrolling-text that appears or disappears at the start and end points in the row region of the frame. Therefore, this method cannot detect scrolling-text that appears or disappears at the start or end points of the frame boundary.

An adaptive temporal differential-based detection method [23] has been proposed to improve the detection accuracy of scrolling-text detection. It calculates the difference between the edge map of the previous and current frames obtained from a Roberts edge detector [13]. Then, the densities of the edge difference map at the four frame boundary regions (top, bottom, right, and left) are calculated to detect the scrolling-text. If the edge density of each frame boundary region is high, the method determines the detected edges as scrolling-text in the frame boundary region. This method provided high performance when the background of the entire horizontal scrolling-text was simple. However, it requires ten previous frames to calculate the edge difference map of the current frame. Moreover, because the edge densities for four frame boundary regions are not high when scrolling-text appears or disappears at the start or end point of the frame boundary, this method cannot detect these types of scrolling-text.

The Histogram Oriented Moments descriptor-based method [24] for scrolling-text detection identifies the direction of the edges and detects scrolling-text. The central concept of this method is that the number of dominant orientations that point towards the centroid of the connected components is larger than the number of dominant orientations that point away from the centroid of the connected components. Furthermore, it uses the Combined Local-Global optical flow method [26] to extract MV information of candidate scrolling-text regions in two consecutive frames. It detects the final scrolling-text regions by comparing the motion direction of the candidate regions with the typical direction in which scrolling-text flows (horizontal or vertical). This method demonstrated improved the performance of the scrolling-text detection compared with previous methods [21]–[23]. However, this method is insufficient for detecting text edges because text edges exist that do not satisfy the new hypothesis used in this method [24]. Therefore, it does not provide robust performance for various types of scrolling-text.

Most recently, a text edge detector method [25] based on the concept of a region-adaptive threshold has been proposed. The central concept is that text edges are more likely to exist in a region with a higher luminance variation. Therefore, this method increases the probability of determining the given pixel as a text edge pixel by setting the threshold value for text edge detection inversely proportional to the luminance variation for a given region. This method detects the text edges in the current frame and the previous frame using a region-adaptive text edge detector. Then, it calculates the difference between the text edge maps of the previous and the current frames and detects the final scrolling-text area using edge density, edge orientation, and MV distribution analysis of the detected text regions. This method successfully detects the scrolling-text that appears or disappears at the start and end points of the frame boundary and the entire horizontal scrolling-text compared with previous methods [21]–[24]. However, the region-adaptive text edge detector can falsely detect highly-textured regions, which have sharp edges, as scrolling-text. Consequently, the accuracy of this method can be further improved.

III. PROPOSED METHOD

The FRUC architecture that uses the proposed scrolling-text detection method consists of four steps (Fig. 2 (a)):

1) RGB-to-YCbCr conversion: FRUC converts the RGB color space of the input frames to the YCbCr color space.
2) ME and MV correction for the scrolling-text region: The FRUC method [15] extracts the Y image (luminance) and calculates MVs information of the current Y image for each block using the previous and current Y images. Then, FRUC uses the proposed scrolling-text detection method and performs MV correction on the scrolling-text regions obtained from the proposed method to make the initial MVs of the detected region to the MVs of the scrolling-text if they are different.
3) MCI: The MCI generates the interpolated frame using the final MVs in the YCbCr color space.
4) YCbCr-to-RGB conversion: The YCbCr color space of the interpolated frame is converted to the RGB color space.

The proposed scrolling-text detection method consists of three steps (Fig. 2 (b)):

1) Generation of the text edge map
2) Generation of the initial scrolling-text map
3) Refinement of the scrolling-text map.

The detailed operations of the proposed method are described in this section.

A. GENERATION OF THE TEXT EDGE MAP

The purpose of scrolling-text detection is to detect the position of text moving horizontally or vertically in the boundary area of the given frame. For a region with text, the spatial variation in luminance of the text regions is larger than in other regions. Furthermore, the text region generally has a dominant orientation of edges. Based on these observations, the text regions can be extracted by analyzing the spatial variation in luminance and the dominant orientation component of the region. However, the method uses a first-order gradient that considers the relationship between only two adjacent pixels, which often detects highly-textured regions as text.

Therefore, we adopt directional coherence to extract regions with a dominant edge orientation and significant spatial change in luminance (Fig. 3). We focus on directional coherence rather than directly using gradient information,

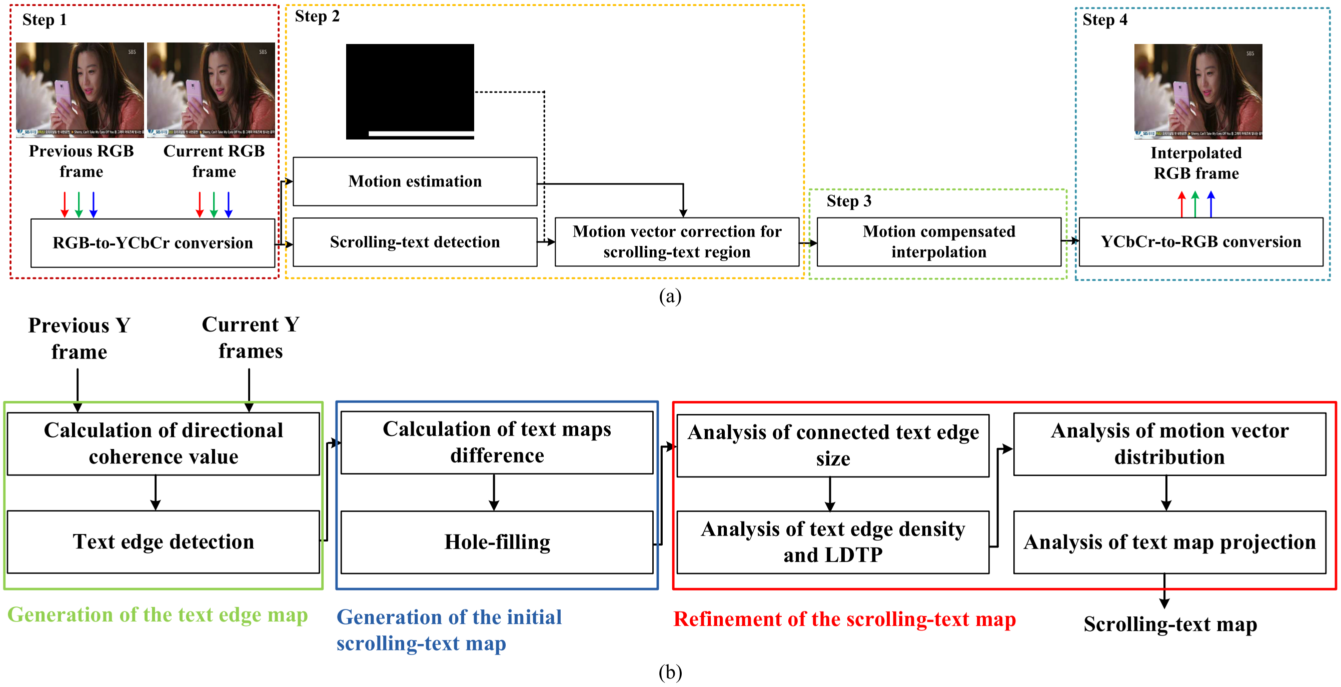

Step 1
Previous RGB frame
Current RGB frame
RGB-to-YCbCr conversion
Step 2
Motion estimation
Scrolling-text detection
Motion vector correction for scrolling-text region
Step 3
Motion compensated interpolation
Step 4
Interpolated RGB frame
YCbCr-to-RGB conversion
(a)
Previous Y frame
Current Y frames
Calculation of directional coherence value
Text edge detection
Calculation of text maps difference
Hole-filling
Analysis of connected text edge size
Analysis of text edge density and LDTP
Analysis of motion vector distribution
Analysis of text map projection
Generation of the text edge map
Generation of the initial scrolling-text map
Refinement of the scrolling-text map
Scrolling-text map
(b)

FIGURE 2. (a) Architecture of FRUC using the proposed scrolling-text detection method and (b) process of the proposed scrolling-text detection method. ©Love from Star SBS.

which is unreliable for highly-textured regions. The patterns of direction generated from the center and surrounding regions can provide a suitable approximation of the underlying image structure, which is coincides with text edges. The proposed method extracts the text edges using a structure tensor, which efficiently summarizes the dominant orientation and the energy along this direction based on the local gradient field, defined as follows:

$$\mathbf{T}_s^k(i) = \begin{bmatrix} \sum_{j\in B_i} I_x^k(j)^2 & \sum_{j\in B_i} I_x^k(j) I_y^k(j) \\ \sum_{j\in B_i} I_x^k(j) I_y^k(j) & \sum_{j\in B_i} I_y^k(j)^2 \end{bmatrix}, \quad (1)$$

where $\mathbf{T}_s^k(i)$ denotes the structure tensor matrix of pixel i at the k-th frame, and I_x^k and I_x^k denote the gradient in the horizontal and vertical directions at the k-th frame, respectively. B_i denotes the neighbor region centered at the i-th pixel position. The block size B_i was set to 5×5 pixels experimentally.

The effectiveness of the structure tensor defined in (1) for our task stems from the fact that the relative discrepancy between two eigenvalues ($\lambda_1 \geq \lambda_2 \geq 0$) of $\mathbf{T}_\mathbf{s}^\mathbf{k}(i)$ indicates how intensively gradients in the local region are distributed along the dominant direction (the degree to which those directions are consistent). Gradients with text edges are strongly distributed along the dominant direction compared with uniform or highly-textured regions (Fig. 3). Therefore, the proposed method defines directional coherence at each pixel position as follows:

$$\xi = (\lambda_1 - \lambda_2). \quad (2)$$

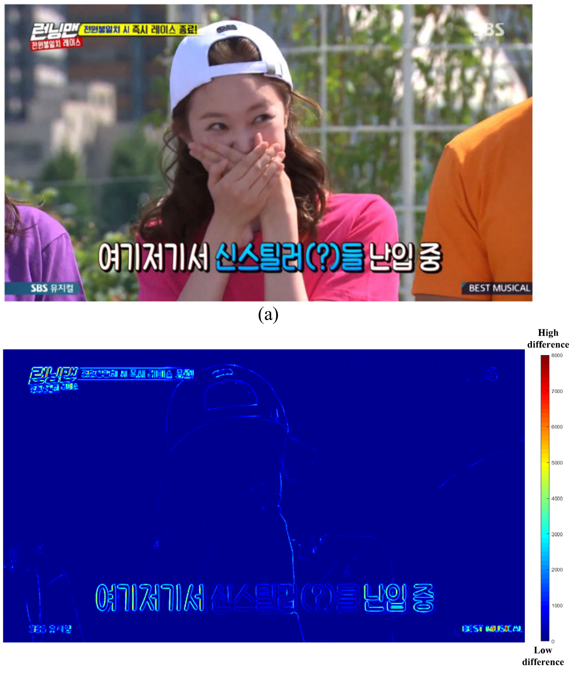
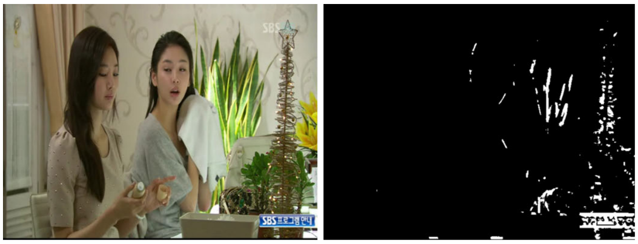


FIGURE 3. Directional coherence values obtained from original image: (a) original image and (b) directional coherence values for each pixel (Note that directional coherence value of the scrolling-text or text region is much larger than that of flat and highly-textured regions). ©Running Man SBS.

The value ξ represents the degree of coherence of the edge directions. Therefore, the larger the ξ value, the higher the directional coherence. If directional coherence for



(a) (b)

FIGURE 4. **Generation of the text map using directional coherence values: (a) original image and (b) text edge map (Black pixels denote non-text regions and white pixels denote text regions). ©Secret Garden SBS.**

(a) (b)

FIGURE 5. **Generation of the initial scrolling-text map: (a) text maps difference using the current and the previous frames and (b) result of hole-filling process. ©Secret Garden SBS.**

the (i, j)-th pixel is larger than a threshold value T_1, we determine the (i, j)-th pixel as the text edge, defined as follows:

$$TM_{i,j} = \begin{cases} 1, & \textit{if } \xi_{i,j} > T_1, \\ 0, & \textit{otherwise}, \end{cases} \tag{3}$$

where $TM_{i,j}$ denotes the text edge for the (i, j)-th pixel, and T_1 denotes the predefined threshold value, set to 1500 experimentally to maximize the F_1 score, which is an evaluation metric used to measure the accuracy of the scrolling-text detection. The T_1 value in (3) is important role to generate a robust text edge map. The proposed method uses the concept of directional coherence to generate a more desirable text edge map while suppressing textured regions in the given frame (Fig. 4).

B. GENERATION OF THE INITIAL SCROLLING-TEXT MAP

The proposed method generates an initial scrolling-text map using the previous and current text maps obtained from the previous step. Before the generation of the initial scrolling-text map, because scrolling-text generally exists within the frame boundary, the result of text map detection in the frame boundary regions is considered to detect the scrolling-text, as in previous papers [21]–[25]. A temporal difference exists between the scrolling-text regions of consecutive frames. The proposed method captures such difference by comparing the previous and current text maps.

Accordingly, we calculate the text map difference between the previous and current frames. By calculating the text map difference, the proposed method can remove the static edge regions and preserve the regions with only moving or scrolling-text pixels. The result of the text edge difference map for the current and previous frames highlights the gaps between two different text edges in the scrolling-text regions (the gaps represent zero points between two text edges in Fig. 5(a)).

The proposed method generates the connected components of the text edges in the result of the text edge difference map by performing a hole-filling process [25], which fills the edges on the gap between two different edges in the result of text edge difference map. In this process, the hole is the gap between two different edges in the same row on the result of the text edge difference map. If the hole is smaller than a predetermined threshold (T_2), we fill the edges with the corresponding gap (Fig. 5 (b)). The threshold value T_2 was set to 32 pixels, as in [25]. The concept of the hole-filling process is to connect the gaps between two different sets of text so that the proposed method can detect consecutive scrolling-text as one block with emphasis on the use of FRUC.

C. REFINEMENT OF THE SCROLLING-TEXT MAP

After generating the initial scrolling-text map, we need to remove the falsely-detected regions using the proposed refinement process. The refinement process consists of the analyses of the connected text edge size, text edge density and LDTP [14], MV distribution, and text map projection.

For the first refinement, the size of edges containing scrolling-text is larger than that of other regions. The proposed method uses this concept for the first refinement process by removing the scrolling-text region if its length is shorter than a predefined length (T_3). The threshold value T_3 was set to 64 empirically by observing the minimum size of scrolling-text region [25].

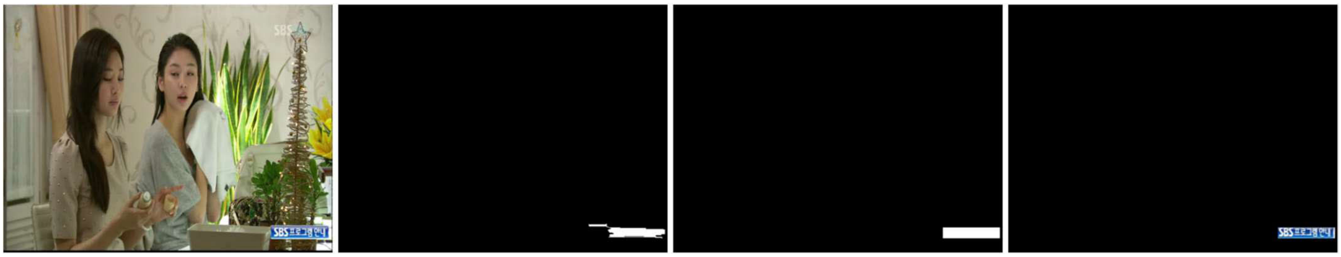
For the second refinement, the scrolling-text region generally has a higher edge density than other regions. Furthermore, it has both various principal directions and a large luminance difference. Based on these observations, we estimate the text edge density and texture patterns. We use LDTP analysis [14] to efficiently describe the texture patterns of the text edges and the luminance variation in the candidate scrolling-text regions. We compute LDTP by calculating the principal directional numbers of the neighborhood using the Kirsch compass masks [27] in eight different directions:

$$P^1_{dir} = \arg\max_i \{C_i | 0 \le i \le 7\}, \tag{4}$$

where P^1_{dir} is the principal directional number, and C_i is the absolute value of the convolution of the image I, with the i-th Kirsch compass mask, M_i, defined as follows:

$$C_i = |I * M_i|, \tag{5}$$

where "$*$" denotes convolution operation of the image with the Kirsch compass mask (filter). Note that "|. |" denotes absolute operation of multiplication between two variables. In [14], the absolute value of the eight Kirsch mask's responses in computed accordingly. The second directional



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

FIGURE 6. **Refinement of the scrolling-text map: (a) original image, (b) first refinement result of the scrolling-text map, (c) final refinement result of the scrolling-text map, and (d) scrolling-text in the detected scrolling-text region. ©Secret Garden SBS.**

number P^2_{dir} is computed in the same way by extracting the second maximum response in (4).

For each principal direction, LDTP calculates the luminance difference between the pixels in the principal direction as follows:

$$D^n_{i,j} = p^n_{i,j} - q^n_{i,j}, \tag{6}$$

where $D^n_{i,j}$ is the n-th difference for the pixel (i, j) in the n-th principal direction, and $p^n_{i,j}$ and $q^n_{i,j}$ are the luminance values of the principal direction and opposite principal direction positions among the eight neighborhood pixels with respect to pixel (i, j), respectively. The method for calculating these local differences is equivalent to that of thresholding in Local Binary Pattern (LBP) [28].

In contrast to the binary coding of LBP, LDTP codes the difference using three levels (negative, equal, and positive). If $D^n_{i,j}$ is larger than a predefined threshold value, ε, the LDTP method encodes a 2 whereas if $D^n_{i,j}$ is smaller than–ε, it encodes a 1. If $D^n_{i,j}$ is between the $-\varepsilon$ and ε values, LDTP encodes a 0. By representing these three levels, LDTP can represent a more distinctive code for the neighborhood. A threshold value ε was set to 15, as in [14].

Next, we compute the number of different LDTPs to consider both principal direction and luminance variation around the text edge pixel. Because LDTP considers Kirsch compass masks [27] in eight different directions and codes luminance difference for two principal directions using three levels, the total number of potentially different LDTPs is $8 \times 3 \times 3 = 72$. The proposed method enlarges the code length by considering the third principal directional and luminance difference for more diverse and potential LDTP representation. The length of the code used in the proposed method is $8 \times 3 \times 3 \times 3 = 216$. Then, we combine the text edge density and diversity degrees of the LDTPs to define the candidate region characteristic for use in the second refinement, as follows:

$$RC_i = TED_i \times LDTP_i, \tag{7}$$

where RC_i, TED_i, and $LDTP_i$ denote the i-th candidate region characteristic, the text edge density, and degree of different LDTPs for the candidate region. The proposed method removes the i-th candidate regions if its RC_i is smaller than the predefined threshold value T_4. The threshold value T_4, was set to 0.08 based on various experimental results.

For the third refinement, the MVs for the scrolling-text region have the same direction as the typical motion of scrolling-text. Based on observation, we can eliminate false detection. The proposed method removes the candidate scrolling-text regions using the results of the MV distribution analysis. The MVs of each candidate region are accumulated in an MV histogram. Then, the proposed method detects the peak value of the MV histogram to represent the MV direction in the candidate region. The proposed method removes the candidate region if the motion direction of peak value in MV histogram for each candidate region is distinctively different from the typical direction of the scrolling-text (horizontal or vertical).

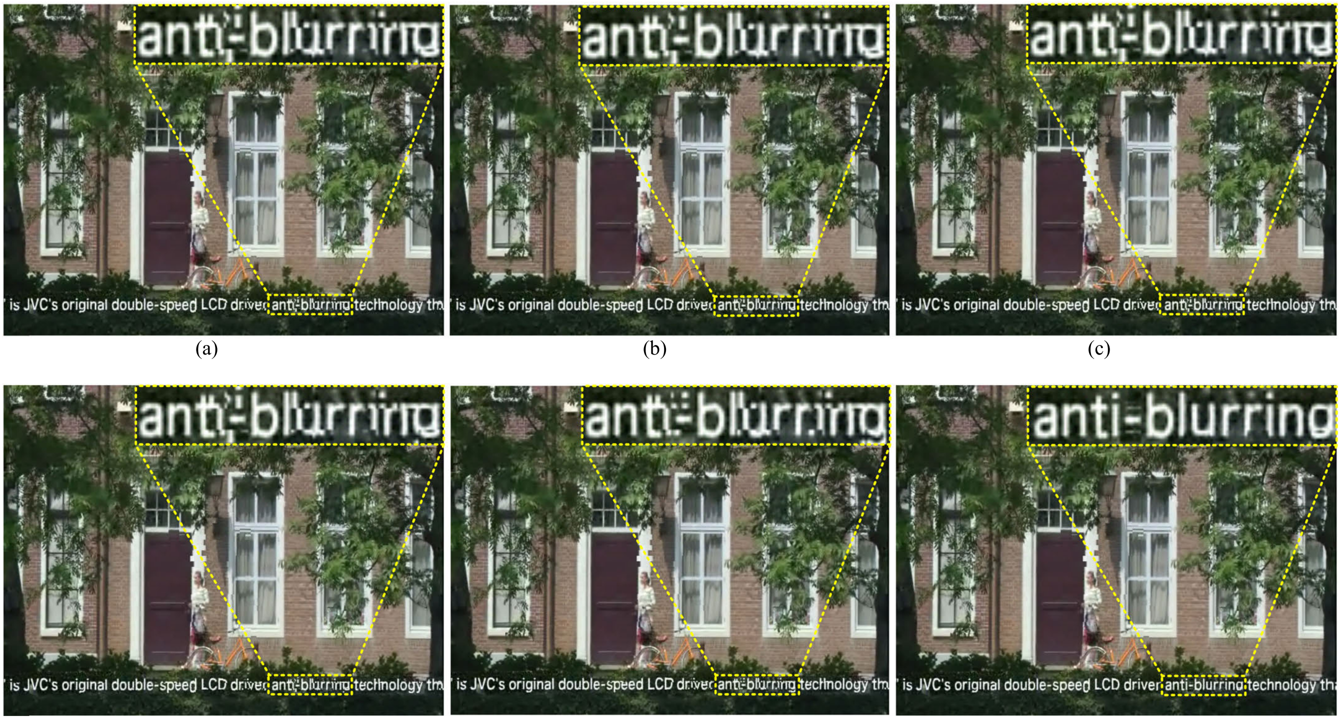
For the fourth refinement, we use a projection of the candidate regions obtained from the third refinement because most of scrolling-text is placed horizontally in the video, the vertically longer candidate regions can be eliminated. The horizontal projection is performed to accumulate all the candidate region pixels in each row to form a histogram of the number of detected pixels. If the number of candidate region pixels among the pixel rows is small, the detected row is removed to refine the detected scrolling-text region. Based on this refinement process, the proposed method can improve the accuracy of the scrolling-text detection further.

Finally, it is reasonable to consider that a scrolling-text area is generally rectangular. Consequently, the proposed method generates each of the remaining candidate areas into the smallest rectangle by linking four points of the remaining candidate regions to contain all candidate regions (Fig. 6).

IV. EXPERIMENTAL RESULTS

We conducted experiments to evaluate the performance of the proposed and the previous scrolling-text detection methods.

First, we visually evaluated the quality of the interpolated frames generated by FRUC using the proposed and the previous scrolling-text detection methods. We focus on developing the scrolling-text method to be used with FRUC. The block size of 8×8 pixels (standard for FRUC applications) and the search range size (a range in which the search can be performed around the current block) of 16 pixels are most widely used in the FRUC applications. Therefore, we set the



(a) (b) (c) (d) (e) (f)

FIGURE 7. Interpolated frames generated using two consecutive frames from JVC sequences generated by: (a) TSTD [21], (b) GSTD [22], (c) HSTD [23], (d) KSTD [24], (e) LSTD [25], and (f) proposed scrolling-text detection method. ©JVC.

block size of the FRUC to 8 × 8 pixels [1], [29], [30] and the search range size of the FRUC to 16 pixels [1], [31], respectively.

Second, we assessed the accuracy of the proposed and the previous scrolling-text detection methods using Precision (P), Recall (R), and F_1 score (F_1) [32]–[35]. These evaluation metrics are defined as follows:

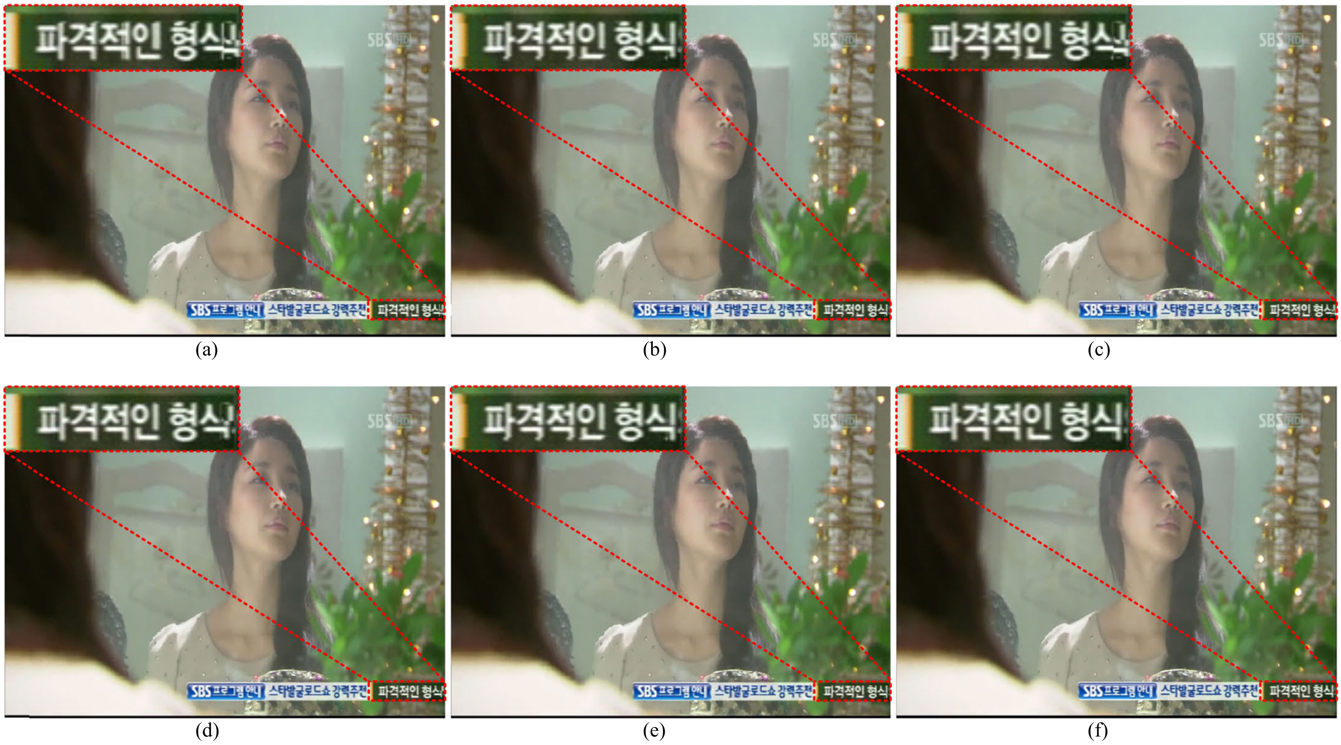
$$P = \frac{Num(D \cap GT)}{Num(D)}, \quad R = \frac{Num(D \cap GT)}{Num(GT)},$$
$$F_1 = 2 \times \frac{P \times R}{P + R}, \tag{8}$$

where D and GT denote a scrolling-text region detected by each method and the ground truth rectangle region of scrolling-text, respectively. Symbol $\cap$ represents the intersection of two groups. $Num(\cdot)$ denotes the number of pixels in a group. We evaluated the performance of scrolling-text detection at the rectangle level to ensure its usefulness with with FRUC techniques. F_1 is an evaluation metric that considers P and R. The range of F_1 is from 0 to 1, where 1 is the best score.

Third, we measured the computation times of the proposed and previous scrolling-text detection methods. As the performance of previous methods to compare the performance of the proposed method, we used five previous methods: Tsai's scrolling-text detection (TSTD) [21], Gim's scrolling-text detection (GSTD) [22], Hsia's scrolling-text detection (HSTD) [23], Khare's scrolling-text detection (KSTD) [24], and Lee's scrolling-text detection (LSTD) [25]. LSTD is the most state-of-the-art scrolling-text detection method.

For all previous methods, we optimized various parameters and set to the values guided by the corresponding papers [21]–[25]. Various parameters used in the proposed method were also optimized to values based on various experiments. For the test sequences, we used video sequences [25] containing various types of scrolling-text: scrolling-text beginning or ending at the frame boundary, scrolling-text that occupies the entire row region of the frame boundary, blurry scrolling-text, and well-distinguished scrolling-text from the background.

In the first experiment, we visually compared the quality of the interpolated frames generated by conventional FRUC [15] using the previous and proposed methods (Figs 7–8). TSTD [21], GSTD [22], HSTD [23], and KSTD [24] could not detect the scrolling-text starting at the frame boundary. Hence, because these methods cannot correct incorrectly-estimated MVs in the scrolling-text regions, they generate severe artifacts at the scrolling-text region in the interpolated frames during FRUC. For LSTD [25], the accuracy of the scrolling-text detection was higher than those of TSTD [21], GSTD [22], HSTD [23], and KSTD [24]. However, when this method was applied to FRUC, it generated artifacts in the scrolling-text regions (Fig. 7 (e), Fig. 8 (e)). In contrast, the proposed method was able to detect various types of the scrolling-text that occupies the entire row region of the frame boundary (Fig. 7) or that begins to scroll from the frame



(a) (b) (c)

(d) (e) (f)

FIGURE 8. Interpolated frames generated using two consecutive frames from Secret Garden 2 sequences generated by: (a) TSTD [21], (b) GSTD [22], (c) HSTD [23], (d) KSTD [24], (e) LSTD [25], and (f) proposed scrolling-text detection method. ©Secret Garden SBS.

TABLE 1. Comparison of the scrolling-text detection accuracy of the proposed and previous methods using precision, recall, and F1 score.

Test sequences (Number of frames)	TSTD [21]			GSTD [22]			HSTD [23]			KSTD [24]			LSTD [25]			Proposed method		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
JVC (390)	0.203	0.511	0.288	0.490	0.950	0.626	0.664	0.799	0.706	0.677	0.769	0.715	0.707	**0.987**	0.821	**0.732**	0.986	**0.838**
High Kick (800)	0.374	0.574	0.434	0.722	0.834	0.751	0.772	0.752	0.714	0.737	0.806	0.763	0.972	0.748	0.841	**0.947**	**0.857**	**0.895**
Secret Garden 1 (600)	0.459	0.504	0.476	0.738	**0.918**	0.801	0.884	0.728	0.785	0.834	0.806	0.816	0.852	0.684	0.840	**0.918**	0.911	**0.908**
Secret Garden 2 (700)	0.235	0.543	0.322	0.644	0.866	0.713	0.822	0.740	0.768	0.727	0.809	0.758	**0.949**	0.894	0.917	0.918	**0.954**	**0.932**
KBS Program 1 (550)	0.438	0.482	0.457	0.853	0.791	0.804	0.738	0.828	0.759	0.851	0.722	0.771	0.887	0.883	0.877	**0.935**	**0.884**	**0.905**
KBS Program 2 (600)	0.507	0.422	0.452	0.628	0.822	0.676	0.720	0.743	0.704	0.722	0.636	0.656	0.849	0.861	0.844	**0.899**	**0.865**	**0.875**
KBS Program 3 (700)	0.580	0.672	0.608	0.805	0.907	0.832	0.822	0.884	0.832	0.770	0.769	0.758	**0.861**	0.924	0.885	0.836	**0.950**	**0.886**
KBS Program 4 (700)	0.551	0.557	0.542	0.952	0.774	0.847	0.969	0.741	0.834	0.909	0.640	0.745	**0.987**	0.782	0.870	0.946	**0.871**	**0.904**
Love From Star (300)	0.339	0.560	0.397	0.497	0.859	0.610	0.734	0.662	0.667	0.755	0.799	0.763	**0.871**	0.829	0.838	0.857	**0.903**	**0.868**
Should We Kiss (370)	0.061	0.192	0.089	0.284	0.911	0.417	0.482	0.971	0.539	0.666	0.773	0.701	0.820	**0.977**	0.886	**0.862**	0.916	**0.887**
Music Video (300)	0.357	0.277	0.306	0.407	**0.898**	0.537	0.575	0.409	0.473	0.890	0.780	0.827	0.952	0.742	0.831	**0.981**	0.829	**0.898**
Discovery of Love (530)	0.231	0.532	0.305	0.428	0.847	0.557	0.731	0.581	0.634	0.757	0.769	0.746	**0.923**	0.786	0.845	0.898	**0.898**	**0.892**
Running Man (650)	0.254	0.494	0.311	0.258	0.982	0.395	0.264	0.979	0.353	0.590	0.686	0.617	0.725	0.997	0.835	**0.758**	**0.998**	**0.860**
Average	0.353	0.486	0.384	0.593	0.874	0.659	0.706	0.755	0.674	0.760	0.751	0.741	0.874	0.867	0.856	**0.883**	**0.909**	**0.888**

Bold, underline values represent the best precision, recall, and F_1 scores in each row.

boundary (Fig. 8). Thus, FRUC with the proposed method can generate high-quality interpolated frames in the scrolling-text regions (Fig. 7 (f), Fig. 8 (f)).

In the second experiment, we evaluated the accuracy of scrolling-text detection of the previous method and the proposed method using P, R, and F_1 [32]–[35]. We counted the number of pixels in the scrolling-text regions and the number of pixels in ground truths for each video sequence and calculated F_1. TSTD [21], GSTD [22], and HSTD [23] use edge or MV information of the image for scrolling-text detection but have difficulty detecting the scrolling-text that appears or disappears at the start or end point of the frame boundary. Therefore, these methods [21]–[23] had a lower F_1 than the other methods [24], [25]. KSTD [24] and

TABLE 2. Average computation time per pixel of the proposed and previous methods.

	TSTD [21]	GSTD [22]	HSTD [23]	KSTD [24]	LSTD [25]	Proposed method
Computation time (C_T [μs])	0.198	2.768	3.485	22.984	9.286	4.413

LSTD [25] were more accurate at detecting scrolling-text and preserving a high F_1 compared with TSTD [21], GSTD [22], and HSTD [23].

The proposed method improved the accuracy of the scrolling-text detection even further when compared with previous methods [24], [25] (Table 1). F_1 of the proposed method was 0.504 (a 131.25% improvement), 0.229 (a 34.78% improvement), 0.210 (a 21.40% improvement), 0.147 (a 19.84% improvement), and 0.032 (a 3.74% improvement) higher than those of TSTD [21], GSTD [22], HSTD [23], KSTD [24], and LSTD [25]. The improvement was calculated by dividing the increment of F_1 (F_1 of the proposed method minus F_1 of the previous method) by the original F_1 for the previous method. This improvement by the proposed method could be attributed to the use of directional coherence values for each pixel in the given image to distinguish text regions from highly-textured regions. Moreover, the proposed method used four types of refinement methods to eliminate non-scrolling-text regions from the initial scrolling-text map. With these refinement processes, the proposed method could accurately detect the start or end of text scrolling at the frame boundary.

In the third experiment, we compared the computation time per pixel (C_T [μs]) of the previous and proposed methods. The proposed method reduced the average C_T by 18.571 μs (an 80.80% reduction) and 4.873 μs (a 52.48% reduction) compared with the KSTD [24] and LSTD [25], respectively (Table 2). TSTD [21], GSTD [22], and HSTD [23] are relatively simple methods, so the average C_T is fast, but the F_1 score is low when compared with that of the proposed method.

V. CONCLUSION

In this article, we proposed a new scrolling-text detection method that uses directional coherence with a scrolling-text-aware refinement technique for FRUC. The central concept of the proposed method is to use the directional coherence of each pixel to distinguish the scrolling-text pixels from textured regions and use refinement processes to detect text appearing and disappearing at the start and end points of the frame boundary.

The proposed method calculates coherence values of edge directions for each pixel to represent the text map. It generates an initial scrolling-text map by calculating the difference between the current and previous text maps. Then, the proposed method refines the initial scrolling-text map by analyzing the texture patterns and MV distribution for the detected text regions. Furthermore, it uses the text map projection to enhance the accuracy of the scrolling-text detection.

The benefits of the proposed method were evaluated in terms of scrolling-text detection accuracy and processing time on the various video sequences.

The experimental results demonstrated that the average F_1 of the proposed method was up to 0.504 (a 131.25% improvement) higher than those of previous methods. The average C_T of the proposed method was up to 18.571 μs lower than those of the previous methods (an 80.80% reduction). Furthermore, FRUC using the proposed scrolling-text detection method could generate the highest-quality interpolated frames for the scrolling-text regions compared with previous methods.

REFERENCES

[1] S.-J. Kang, S. Yoo, and Y. H. Kim, "Dual motion estimation for frame rate up-conversion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, pp. 1909–1914, Dec. 2010.

[2] S.-J. Kang, K.-R. Cho, and Y. Kim, "Motion compensated frame rate up-conversion using extended bilateral motion estimation," *IEEE Trans. Consum. Electron.*, vol. 53, no. 4, pp. 1759–1767, Nov. 2007.

[3] G. Dane and T. Q. Nguyen, "Motion vector processing for frame rate up conversion," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 3, May 2004, pp. 309–312.

[4] S.-J. Kang, D.-G. Yoo, S.-K. Lee, and Y. Kim, "Multiframe-based bilateral motion estimation with emphasis on stationary caption processing for frame rate up-conversion," *IEEE Trans. Consum. Electron.*, vol. 54, no. 4, pp. 1830–1838, Nov. 2008.

[5] A. J. Patti, M. I. Sezan, and A. M. Tekalp, "High resolution standards conversion of low resolution video," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, vol. 4, May 1995, pp. 2197–2200.

[6] K. Sekiya and H. Nakamura, "Eye-trace integration effect on the perception of moving pictures and a new possibility for reducing blur on hold-type displays," in *SID Symp. Dig. Tech. Papers*, vol. 33, no. 1, May 2002, pp. 9309–9333.

[7] K. Hilman, H.-W. Park, and Y.-M. Kim, "Using motion-compensated frame-rate conversion for the correction of 3: 2 pulldown artifacts in video sequences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 6, pp. 8969–8977, Sep. 2000.

[8] S. H. Abdulhussain, A. R. Ramli, A. J. Hussain, B. M. Mahmmod, and W. A. Jassim, "Orthogonal polynomial embedded image kernel," in *Proc. Int. Conf. Inf. Commun. Technol. (ICICT)*, New York, NY, USA, 2019, pp. 215–221.

[9] Q. Ying-Dong, C. Cheng-Song, C. San-Ben, and L. Jin-Quan, "A fast subpixel edge detection method using Sobel–Zernike moments operator," *Image Vis. Comput.*, vol. 23, no. 1, pp. 11–17, Jan. 2005.

[10] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[11] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Alvey Vis. Conf.*, 1988, pp. 147–151.

[12] I. E. Sobel, "Camera models and machine perception," Stanford Univ. Artif. Intell. Lab, Stanford, CA, USA, Tech. Rep. AIM-21, 1970.

[13] L. G. Roberts, "Machine perception of three-dimensional solids," in *Optical and Electro-Optical Information Processing*. Cambridge, MA, USA: MIT Press, 1965, pp. 159–197.

[14] A. Ramírez Rivera, J. Rojas Castillo, and O. Chae, "Local directional texture pattern image descriptor," *Pattern Recognit. Lett.*, vol. 51, pp. 94–100, Jan. 2015.

[15] B.-D. Choi, J.-W. Han, C.-S. Kim, and S.-J. Ko, "Frame rate up-conversion using perspective transform," *IEEE Trans. Consum. Electron.*, vol. 52, no. 3, pp. 975–982, Aug. 2006.

[16] L. Wu, P. Shivakumara, T. Lu, and C. L. Tan, "A new technique for multi-oriented scene text line detection and tracking in video," *IEEE Trans. Multimedia*, vol. 17, no. 8, pp. 1137–1152, Aug. 2015.

[17] S. Dey, P. Shivakumara, K. S. Raghunandan, U. Pal, T. Lu, G. H. Kumar, and C. S. Chan, "Script independent approach for multi-oriented text detection in scene image," *Neurocomputing*, vol. 242, pp. 96–112, Jun. 2017.

[18] A. Mittal, P. P. Roy, P. Singh, and B. Raman, ''Rotation and script independent text detection from video frames using sub pixel mapping,'' *J. Vis. Commun. Image Represent.*, vol. 46, pp. 187–198, Jul. 2017.

[19] P. Shivakumara, L. Wu, T. Lu, C. L. Tan, M. Blumenstein, and B. S. Anami, ''Fractals based multi-oriented text detection system for recognition in mobile video images,'' *Pattern Recognit.*, vol. 68, pp. 158–174, Aug. 2017.

[20] A. Sain, A. K. Bhunia, P. P. Roy, and U. Pal, ''Multi-oriented text detection and verification in video frames and scene images,'' *Neurocomputing*, vol. 275, pp. 1531–1549, Jan. 2018.

[21] T. H. Tsai, Y. C. Chen, and C. L. Fang, ''2DVTE: A two-directional videotext extractor for rapid and elaborate design,'' *Pattern Recognit.*, vol. 42, no. 7, pp. 1496–1510, Jul. 2009.

[22] G. Y. Gim, Y. J. Kim, T.-G. Ahn, and S.-H. Park, ''Horizontal scrolling text processing for frame rate conversion,'' in *Proc. IEEE 2nd Int. Conf. Consum. Electron. Berlin (ICCE-Berlin)*, Sep. 2012, pp. 333–3309.

[23] S.-C. Hsia and N.-T. Chang-Jian, ''Efficient scrolling videotext detection with adaptive temporal differential approach,'' *IET Image Process.*, vol. 8, no. 8, pp. 455–463, Aug. 2014.

[24] V. Khare, P. Shivakumara, and P. Raveendran, ''A new histogram oriented moments descriptor for multi-oriented moving text detection in video,'' *Expert Syst. Appl.*, vol. 42, no. 21, pp. 7627–7640, Nov. 2015.

[25] H. S. Lee, S.-J. Kang, and Y. H. Kim, ''Scrolling text detection based on region characteristic analysis for frame rate up-conversion,'' *Displays*, vol. 55, pp. 19–30, Dec. 2018.

[26] A. Bruhn, J. Weickert, and C. Schnrr, ''Lucas/kanade meets horn/schunk: Combining local and global optical flow methods,'' *Int. J. Comput. Vis.*, vol. 61, no. 3, pp. 211–231, Feb. 2005.

[27] R. A. Kirsch, ''Computer determination of the constituent structure of biological images,'' *Comput. Biomed. Res.*, vol. 4, no. 3, pp. 315–328, Jun. 1971.

[28] C. Shan, S. Gong, and P. W. McOwan, ''Facial expression recognition based on local binary patterns: A comprehensive study,'' *Image Vis. Comput.*, vol. 27, no. 6, pp. 803–816, May 2009.

[29] U. S. Kim and M. H. Sunwoo, ''New frame rate up-conversion algorithms with low computational complexity,'' *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 3, pp. 384–393, Mar. 2014.

[30] N. Jacobson, Y.-L. Lee, V. Mahadevan, N. Vasconcelos, and T. Q. Nguyen, ''A novel approach to FRUC using discriminant saliency and frame segmentation,'' *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2924–2934, Nov. 2010.

[31] D.-G. Yoo, S.-J. Kang, and Y. Hwan Kim, ''Direction-select motion estimation for motion-compensated frame rate up-conversion,'' *J. Display Technol.*, vol. 9, no. 10, pp. 840–850, Oct. 2013.

[32] Y. Yang and X. Liu, ''A re-examination of text categorization methods,'' in *Proc. 22nd Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 1999, pp. 42–49.

[33] S.-J. Kang, S. I. Cho, S. Yoo, and Y. H. Kim, ''Scene change detection using multiple histograms for motion-compensated frame rate up-conversion,'' *J. Display Technol.*, vol. 8, no. 3, pp. 121–126, Mar. 2012.

[34] S. I. Cho and S.-J. Kang, ''Histogram shape-based scene-change detection algorithm,'' *IEEE Access*, vol. 7, pp. 27662–27667, Feb. 2019.

[35] S. H. Abdulhussain, A. R. Ramli, B. M. Mahmmod, M. I. Saripan, S. A. R. Al-Haddad, T. Baker, W. N. Flayyih, and W. A. Jassim, ''A fast feature extraction algorithm for image and video processing,'' in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.

HO SUB LEE (Member, IEEE) received the B.S. degree in electrical and electronic engineering from Kyungpook National University, South Korea, in 2014, and the M.S. and Ph.D. degrees in electrical and electronic engineering from the Pohang University of Science and Technology, in 2016 and 2020, respectively. He is currently an Assistant Professor of electronic engineering with Kyungnam University, South Korea. His current research interests include image analysis, computer vision, and circuit design for display and multimedia systems.

SUNG IN CHO (Member, IEEE) received the B.S. degree in electronic engineering from Sogang University, South Korea, in 2010, and the Ph.D. degree in electrical and computer engineering from the Pohang University of Science and Technology, in 2015. From 2015 to 2017, he was a Senior Researcher with LG Display. From 2017 to 2019, he was an Assistant Professor of electronic engineering with Daegu University. He is currently an Assistant Professor of multimedia engineering with Dongguk University, Seoul. His current research interests include image analysis and enhancement, video processing, multimedia signal processing, and circuit design for LCD and OLED systems.