IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 30, 2021, accepted December 27, 2021, date of publication January 27, 2022, date of current version February 14, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3146645

Luminance Level of Histogram-Based Scene-Change Detection for Frame Rate Up-Conversion

HO SUB LEE[1], (Member, IEEE), AND SUNG IN CHO[2], (Member, IEEE)

[1]Department of Electronics and Electrical Engineering, Daegu University, Gyeongsan-si 38453, Republic of Korea
[2]Department of Multimedia Engineering, Dongguk University, Seoul 04620, Republic of Korea

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

This work was supported by the National Research Foundation of Korea (NRF) funded by the Korean Government (MSIT; Ministry of Science, ICT & Future Planning) under Grant 2020R1G1A1102163.

ABSTRACT Scene change detection is an essential process of frame rate up-conversion (FRUC). The performance of FRUC highly dependents on the accuracy of scene change detection. This paper proposes a new scene-change detection method that uses analysis of luminance level of the histograms for FRUC. The histogram luminance level refers to the statistical average luminance value obtained from the generated histograms for each region. Existing histogram-based scene change methods calculate the difference between optimal threshold values using an automatic thresholding technique or extract the difference between the histogram shape to detect the scene change. The automatic thresholding method uses iterative operations—the difference between the histogram shape is simply a method of calculating the luminance difference for the current and previous frames. Thus, it requires many computational resources and incorrectly detects a scene change because calculating the histogram shape cannot reflect regional image characteristics. The proposed method addresses these problems using histogram luminance levels for each region in the given frames. It calculates the level differences between the previous and current frames to detect the initial scene change regions. Moreover, the proposed method refines the initial scene change regions by analyzing the distribution of surrounding detected regions and uses refinement to enhance scene-change detection accuracy. In the experimental results, the proposed method increased the average F1 score to 0.4816 (a 122.51% improvement) compared with the benchmark methods. The average computation time per pixel of the proposed method also decreased to 13.5323 μs (a 87.06% reduction) compared with the benchmark methods.

INDEX TERMS Frame rate up-conversion, histogram luminance level, scene-change detection, motion estimation.

I. INTRODUCTION

Frame rate up-conversion (FRUC) is a technique that increases the frame rate of original videos by inserting interpolated frames between two consecutive frames [1]–[4]. Interpolated frames are generated using motion vectors (MVs), which is the displacement of an object between consecutive frames. FRUC has been used for various applications, including film-to-video conversion to increase the frame rate of films at 24 frames per second (fps) to 50 or 60 fps [5], motion blur reduction in hold-type displays such as liquid crystal displays (LCDs) [6], and TV standard conversion with different frame rates [7]. FRUC is an essential technique to match the fps of the input video and display system.

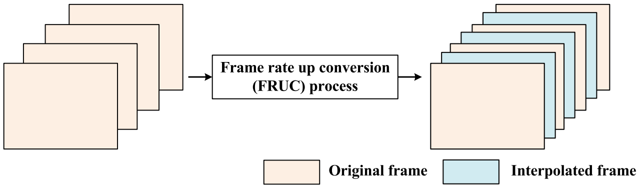


FIGURE 1. Basic illustration of FRUC based on the frame interpolation scheme.

FRUC consists of three primary steps [1]–[4]: motion estimation (ME), MV smoothing (MVS), and motion-compensated interpolation (MCI). ME calculates MVs of

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad E. H. Chowdhury.

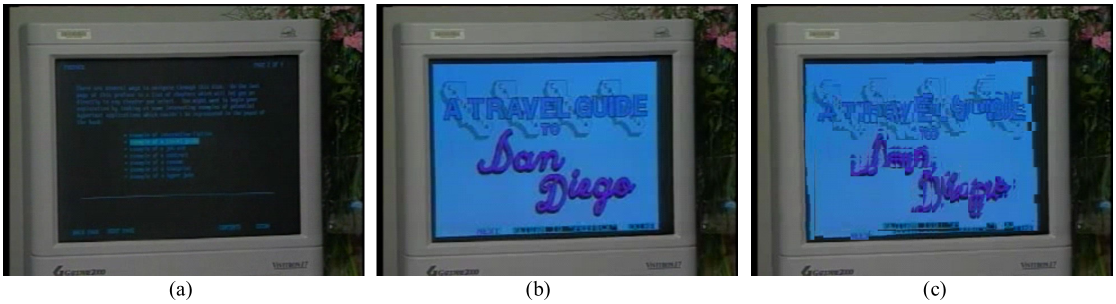


FIGURE 2. Comparison of interpolated frames: (a) previous frame, (b) current frame, and (c) interpolated frame without scene-change detection.

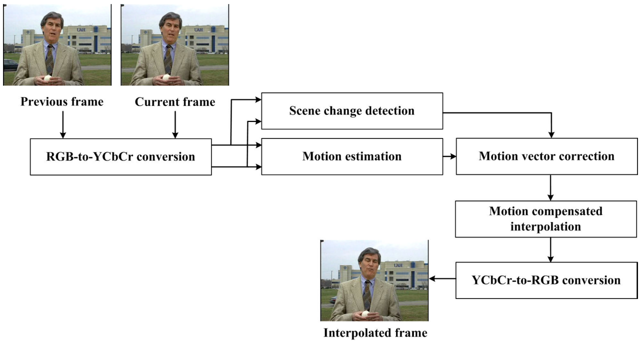


FIGURE 3. Overall architecture of frame rate up-conversion (FRUC).

an object, a displacement between two consecutive frames. MVS corrects the outliers in a set of MVs called by an MV field. MCI generates a new interpolated frame between two original frames using the calculated MVs, as depicted in Fig. 1.

ME is the most important of these steps because the performance of FRUC is highly dependent on the accuracy of the MVs calculated by the ME, assuming that the same object exists in the consecutive frames. Conventional FRUC often fails to estimate MVs accurately when an object does not exist within the consecutive frames due to a scene change, so the interpolated frames with poor quality are generated, as depicted in Fig. 2. Therefore, the scene-change detection process—which considers both a local scene change, a partial change between consecutive frames, and a global scene change, which is the entire change between consecutive frames—is essential for FRUC to correct the MVs to improve the quality of interpolated frames.

For decades, several scene-change detection methods have been proposed. The MV-based scene-change detection method [5] calculates the block-based MV difference between the previous and current frames to detect the scene change. Motion residual ratio-based scene-change detection [6] was proposed to increased scene-change detection accuracy. This method uses the ratio of the sum of the absolute difference MVs and motion residuals between the previous and current frames to detect the scene change.

An optical flow-based scene-change detection method was proposed [7] to improve scene-change detection accuracy. This method uses the statistical properties of optical flows—a valuable and effective method to track the motion of an object between consecutive frames—to find the MVs in the consecutive video frames. The optical flow-based scene-change detection method can detect the rapid scene changes in the video frames by analyzing the variation in optical flows.

The histogram-based scene-change detection method detects a scene change using the difference between the histogram shape between the previous and current frames. The conventional scene-change detection methods described previously only considered a global scene change between consecutive frames. However, the local scene change should be considered to improve the quality of an interpolated frame by correcting the wrongly calculated MV in the local scene change region during the FRUC process.

Therefore, a block-based histogram-based method [8] was proposed to consider local scene change for FRUC application. In this method, a histogram of the block region corresponding to each consecutive frame was generated, and the difference between the block-based histograms was calculated to detect the local scene change. However, because this method calculates the difference between the histogram shape of the corresponding regions for the current and the previous frames, it is difficult to calculate an accurate difference value between the two regions.

Consequently, a scene-change detection method [9] that can automatically determine the threshold value was proposed to solve this problem. It generates the histograms for each block between consecutive frames and analyzes the histogram distribution using the Otsu method [10] for automatic thresholding to determine the scene change. However, this method requires significant computation because iterative operations are required to calculate the threshold.

A histogram shape-based scene-change detection method [11] was proposed. This method extracts the shape of the histogram for previous and current frames. It then determines the scene change for each block by calculating the point-based distance using the extracted histogram shape. This method also uses block merging and block smoothing to improve scene-change detection accuracy. This method accumulates the difference between the luminance values corresponding to the same position of the current frame and the previous frame, so it is vulnerable to fast-moving objects, camera motion, and image noise components. Accordingly, there is an opportunity to improve the performance of scene-change detection by accurately extracting the region characteristics of the corresponding frames.

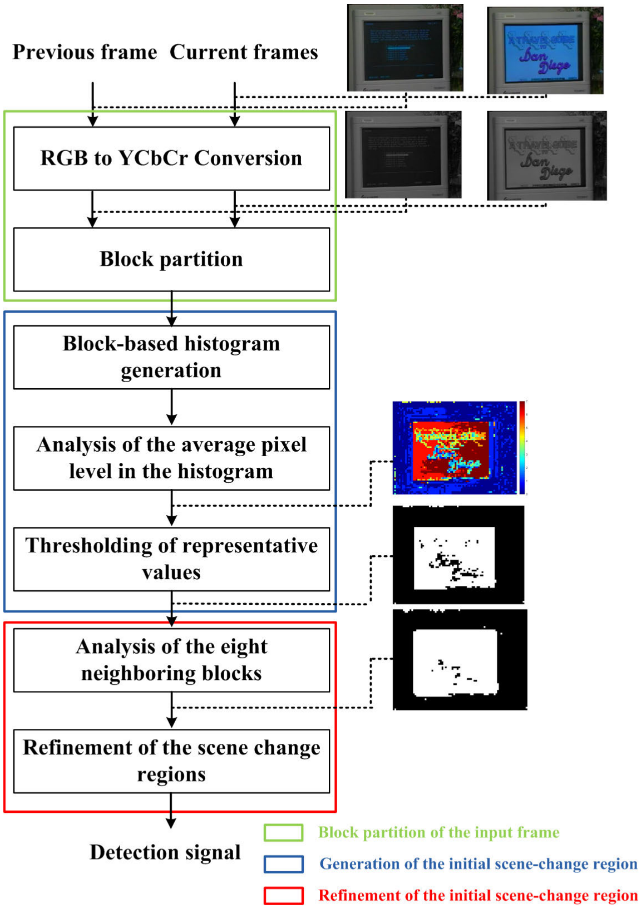

Previous frame
Current frames
RGB to YCbCr Conversion
Block partition
Block-based histogram generation
Analysis of the average pixel level in the histogram
Thresholding of representative values
Analysis of the eight neighboring blocks
Refinement of the scene change regions
Detection signal
🟩 Block partition of the input frame
🟦 Generation of the initial scene-change region
🟥 Refinement of the initial scene-change region

FIGURE 4. **Block diagram of the proposed scene-change detection method and result images generated by each proposed step.**

Most recently, owing to the great success of the deep neural network (DNN), in the field of computer vision task, scene change detection methods using DNN have also been proposed. A deep graph matching network-based scene change detection method that establishes object correspondence between an image pair is proposed [12]. This method can detect object-wise scene changes using the object matching technique without precise image alignment. For object detection, they use Faster R-CNN [13] with ResNet-101 [14] to define the bounding boxes for each of the input images. For determination of scene changes, scene classification-based method was proposed [15]. Two convolutional modules are used to extract the deep representation of bi-temporal input frames. CorrFusion module consists of fully connected layer and batch normalization which are employed to project bi-temporal features into a lower dimensional feature space and normalize the features. To obtain the final scene classification, this method uses the softmax layers. To detect the scene change in a target domain using prior knowledge learned from multiple source domains, a selective adversarial adaptation-based scene change detection method was proposed [16]. The adaptation between multisource and target domains is performed by two domain discriminators. Adversarial learning is used to align the distribution of selected source and target samples. A classification method by adding depth information to assist the sematic information for better detecting scene changes has been proposed [17]. In this method, the depth information can be expressed more accurately by designing the modification strategy which combines high-level sematic features and low-level edge sensitive features. However, since the above mentioned methods are DNN-based methods which require training process and many computation resources for competitive performance, they are not suitable for FRUC application. In this paper, we focus on the development of the scene change detection suitable for FRUC application that does not require a large amount of computation resources and any training process.

In this paper, we propose extracting the luminance level in the histogram-based scene-change detection method for FRUC. The proposed method uses a new operation module based on the previous histogram-based scene change methods [8]–[11] to improve the performance of the scene change. We considered the statistical characteristics to reflect the corresponding region of the given frames more efficiently than the existing histogram-based scene change methods [8]–[11]. In addition, the proposed method further improves the detection accuracy compared to the existing histogram-based scene change methods [8]–[11] by combining a post-processing that refines the initial scene change regions. Specifically, it generates a histogram of each block for previous and current frames. The proposed method then analyzes the statistical characteristics of the generated histogram to extract the luminance level of each region between the previous and current frames. It generates the block-level histograms and extracts the luminance level by removing the meaningless pixel information in the histogram to extract the luminance level more accurately than calculating the average luminance value for each block.

We determine that a scene change occurs by calculating the difference between the extracted luminance levels in the given consecutive frames for the corresponding region. Furthermore, the proposed method uses the correction method of the falsely detected regions to increase scene-change detection accuracy. The main contributions of this paper are as follows:

1) The proposed method uses the histogram luminance level to extract the luminance characteristics for each block of the frames—a simple yet effective method to extract the characteristics of the corresponding region.
2) The proposed method requires minimal computation compared to other existing scene-change detection methods because it does not require an iterative operation. We also use a refinement method to correct the

initial scene change regions detected using the difference in luminance levels in the previous step to improve scene-change detection accuracy.

3) We verified the performance of the proposed scene-change detection method by comparing the interpolated frames generated by the conventional FRUC algorithm [18]. Moreover, we compared the overall performance of the proposed algorithm by identifying the relationship between the scene-change detection accuracy and the processing time.

The remainder of this paper is organized as follows. Section II describes the proposed scene-change detection method. Section III compares the scene-change detection accuracy of benchmark methods with that of the proposed method. Furthermore, Section III presents the quality evaluation of the interpolated frames generated by conventional FRUC and the computational complexity of the proposed and benchmark methods. Finally, Section IV concludes the paper.

II. PROPOSED METHOD

The overall FRUC architecture that uses the proposed scene-change detection method consists of four steps (Fig. 3): RGB-to-YCbCr conversion, ME and MV correction, MCI, and YCbCr-to-RGB conversion. In Step 1, FRUC converts the RGB color space to the YCbCr color space for input frames. In Step 2, the FRUC method extracts the Y image (i.e., the luminance). It then calculates MV information of the current Y image for each block using the previous and current Y images. FRUC generates the interpolated frame using the result of the proposed scene-change detection method. When a scene change is detected, FRUC is not applied to the first frame after the scene change to prevent the artifacts in the interpolated frame caused by incorrect MVs.

If the FRUC process detects the global scene change, it replaces the interpolated frame with the previous frame, and if it detects the local scene change, the FRUC repeats the previous frame only for the scene change regions to prevent visual artifacts of the interpolated frame. MV correction modifies erroneously estimated MV information. In Step 3, the MCI generates the interpolated frame in the YCbCr color space using the modified MV information. In Step 4, the FRUC method generates interpolated frame in the YCbCr color space and converts it to the RGB color space for the resultant frame.

The proposed scene-change detection method consists of three steps (Fig. 4): block partition of the input frames, generation of the initial scene change region, and refinement of the initial scene-change region. The detailed operations of the proposed scene-change detection method are described in detail as follows.

A. BLOCK PARTITION OF THE INPUT FRAMES

First, the proposed method converts the RGB color space of input frames into the YCbCr color space. It then extracts only the Y component among the YCbCr components to generate the luminance image. The purpose of the proposed method is to detect both global and local scene changes. Therefore, we divide the input frames into several blocks of 8 × 8 pixels to consider both global and local scene changes. This block size is the most widely used in FRUC applications [1], [19], [20]. The block-based histogram generation is then performed for each previous and current block in the next step.

B. GENERATION OF THE INITIAL SCENE-CHANGE REGION

The central idea of the proposed method is that the difference between the histogram luminance levels for the corresponding regions of the previous and current frames is significant. Hence, the histogram luminance level differs remarkably when a scene change occurs. The proposed method analyzes the block-based histogram to detect a local scene change. If the scene change occurs locally, the proposed method detects the scene change using the difference between the region characteristics (i.e., the difference in luminance levels).

After generating the histogram for each region, the proposed method removes the meaningless values for pixels that occupy less than 10% of the total number of pixels. We analyze the luminance level in the histogram by determining the representative values of the generated histograms as follows:

$$\begin{aligned} AL_{i,j} &= \sum_{k=1}^{n} k \cdot \left(BH_{i,j}(k)/T_{i,j}\right), \\ T_{i,j} &= \sum_{k=1}^{n} BH_{i,j}(k), \end{aligned} \tag{1}$$

where $BH_{i,j}(k)$ is the histogram value of the (i, j)-th block, k is the bin index of the generated histogram, $T_{i,j}$ is the total histogram value of the (i, j)-th block-based, and n is the total number of bins (set to 26 based on extensive experimental results). After extracting the representative values of the histograms, we calculate the difference between the representative values for each corresponding block for the previous and current frames as follows:

$$D^{P}_{i,j} = \left|AL^{prev}_{i,j} - AL^{curr}_{i,j}\right|, \tag{2}$$

where $AL^{prev}_{i,j}$ and $AL^{curr}_{i,j}$ are the luminance level of the (i, j)-th previous and current blocks, respectively, and $D^{P}_{i,j}$ is the difference in luminance levels between the (i, j)-th previous and current blocks.

Then, the proposed method determines the initial scene change regions through thresholding of the luminance level difference using a predetermined threshold. In determining the initial scene change regions, if the difference in luminance levels for each block of the previous and current frames is greater than a predefined threshold (T_1), the corresponding region is regarded as the initial scene change region. This process can be defined as follows:

$$IB_{i,j} = \begin{cases} 1, & if\ D^{P}_{i,j} > T_1, \\ 0, & otherwise, \end{cases} \tag{3}$$

where $IB_{i,j}$ is the (i, j)-th initial block where the scene change occurs and T_1 is the predefined threshold value (set to 5 based on various experimental results).

C. REFINEMENT OF THE INITIAL SCENE-CHANGE REGION

After generating the initial change regions, we need to refine the detected regions through the proposed refinement process. This process determines the scene change regions by analyzing whether a scene change occurs in the neighboring blocks surrounding the current block. We divide the eight neighboring blocks into four regions: top, bottom, right, and left.

The top region ($R^T_{i,j}$) includes $IB_{i,j-1}$, $IB_{i-1,j-1}$, $IB_{i-1,j}$, $IB_{i-1,j+1}$, and $IB_{i,j+1}$ and the bottom region ($R^B_{i,j}$) includes $IB_{i,j-1}$, $IB_{i+1,j-1}$, $IB_{i+1,j}$, $IB_{i,j-1}$, and $IB_{i+1,j+1}$. The right region ($R^R_{i,j}$) includes $IB_{i-1,j}$, $IB_{i-1,j+1}$, $IB_{i,j+1}$, $IB_{i+1,j+1}$, and $IB_{i+1,j}$ and the left region ($R^L_{i,j}$) includes $IB_{i-1,j}$, $IB_{i-1,j-1}$, $IB_{i,j-1}$, $IB_{i+1,j-1}$, and $IB_{i+1,j}$. After dividing each block into four regions, the proposed method finally determines the corresponding region as the scene change block if the number of initial scene change blocks is greater than a predetermined threshold (T_2) as follows:

$$
\begin{aligned}
R^T_{i,j} &= \frac{1}{5}\left(IB_{i,j-1} + IB_{i-1,j-1} + IB_{i-1,j}\right.\\
&\quad \left. + IB_{i-1,j+1} + IB_{i,j+1}\right),\\
R^B_{i,j} &= \frac{1}{5}\left(IB_{i,j-1} + IB_{i+1,j-1} + IB_{i+1,j}\right.\\
&\quad \left. + IB_{i,j-1} + IB_{i+1,j+1}\right),\\
R^R_{i,j} &= \frac{1}{5}\left(IB_{i-1,j} + IB_{i-1,j+1} + IB_{i,j+1}\right.\\
&\quad \left. + IB_{i+1,j+1} + IB_{i+1,j}\right),\\
R^T_{i,j} &= \frac{1}{5}\left(IB_{i-1,j} + IB_{i-1,j-1} + IB_{i,j-1}\right.\\
&\quad \left. + IB_{i+1,j-1} + IB_{i+1,j}\right),\\
FB_{i,j} &= \begin{cases} 1, & if\ (R^T_{i,j} or R^B_{i,j} or R^R_{i,j} or R^T_{i,j}) > T_2,\\ 0, & otherwise, \end{cases}
\end{aligned}
\tag{4}
$$

where $FB_{i,j}$ is the (i, j)-th final block where the scene change occurs and T_2 is the predefined threshold value to determine the final scene change region. T_2 is set to 0.5 in this paper.

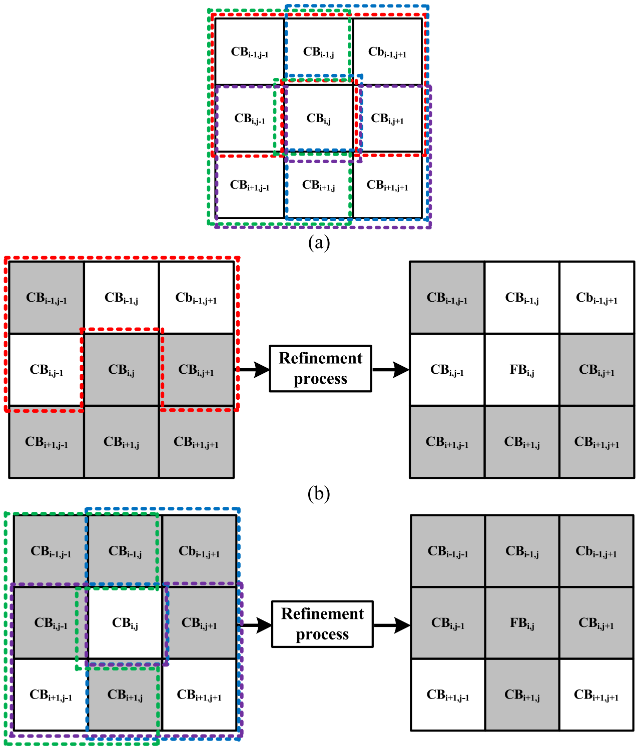
The central idea for refinement is that even if the current block is not designated as the initial scene change region, the possibility of the scene change for the current block is high if the neighboring blocks surrounding the current block contain scene changes (Fig. 5). Similarly, even if the current block is designated as the initial scene change region, the possibility of a scene change for the current block is low if the neighboring blocks surrounding the current block does not contain scene changes. It is reasonable to consider splitting the eight neighboring blocks into four portions directionally. With this refinement process, the proposed method can accurately detect the scene-change regions.

III. EXPERIMENTAL RESULTS

We performed various experiments to evaluate the performance of the proposed and benchmark scene-change detection methods. First, we visually compared the quality of the interpolated frames generated by the FRUC using the proposed and benchmark scene-change detection methods. We set the block size and search range size of the FRUC to 8 × 8 pixels [1], [19], [20] (which is most widely used for the FRUC applications) and 16 pixels [1], [21], respectively. The search range size is a range that the search can be performed around the current block.

Second, we evaluated scene-change detection accuracy for the proposed and benchmark methods. We evaluated scene-change detection accuracy using the most popular evaluation metrics: Precision (P), Recall (R), and F$_1$ score (F$_1$) [9], [11], [22]–[24]. These evaluation metrics are defined as follows:

$$
P = \frac{N_{TP}}{N_{TP} + N_{FP}}, \quad R = \frac{N_{TP}}{N_{TP} + N_{FN}}, \quad F_1 = 2 \times \frac{P \times R}{P + R},
\tag{5}
$$



(a)

(b)

(c)

FIGURE 5. (a) Four surrounding regions used in refinement process of the proposed scene-change detection method, (b) an example of the case where the current block is finally determined as the scene change block after refinement, and (c) an example of the case in which the current block is finally determined as non-scene change block after refinement.

where N_{TP}, N_{FP}, and N_{FN} are the number of correctly detected as a scene change (true positives), the number of incorrectly detected as a scene change (false positives), and the number of incorrectly detected as not a scene change (false negatives), respectively. The F$_1$ score considers both

TABLE 1. Comparison of the scene change detection accuracy of the proposed and the previous methods using precision, recall, and F1 score.

Test sequences (# of total frames)	Scene-change detection method	Threshold	# of scene change frames	Factors for scene-change detection rate					
				NTP	NFP	NFN	P	R	F1
Four Forces Of Flight (600)	BMSCD	100000	78	68	10	250	0.8718	0.2138	0.3434
	OFSCD	10		62	16	69	0.7949	0.4733	0.5933
	ATSCD	d=3, l=0.5		71	7	19	0.9103	0.7889	0.8452
	DMSVD	0.5		34	44	2	0.4359	0.9444	0.5965
	HSSCD	d=200, l=0.5		44	34	9	0.5641	0.8302	0.6718
	Proposed SCD	T_1=5, T_2=0.5		66	12	15	0.8462	0.8148	0.8302
Future Of Transportation (250)	BMSCD	100000	19	12	7	35	0.6316	0.2553	0.3636
	OFSCD	10		16	3	14	0.8421	0.5333	0.6531
	ATSCD	d=3, l=0.5		14	5	0	0.7368	1.0000	0.8485
	DMSVD	0.5		14	5	0	0.7368	1.0000	0.8485
	HSSCD	d=200, l=0.5		15	4	2	0.7895	0.8824	0.8333
	Proposed SCD	T_1=5, T_2=0.5		17	2	5	0.8947	0.7727	0.8293
Composited and Testing Questions (500)	BMSCD	100000	16	16	0	60	1.0000	0.2105	0.3478
	OFSCD	10		16	0	5	1.0000	0.7619	0.8649
	ATSCD	d=3, l=0.5		16	0	1	1.0000	0.9412	0.9697
	DMSVD	0.5		14	2	0	0.8750	1.0000	0.9333
	HSSCD	d=200, l=0.5		13	3	0	0.8125	1.0000	0.8966
	Proposed SCD	T_1=5, T_2=0.5		15	1	0	0.9375	1.0000	0.9677
NASA Vision (450)	BMSCD	100000	12	12	0	33	1.0000	0.2667	0.4211
	OFSCD	10		5	7	1	0.4167	0.8333	0.5556
	ATSCD	d=3, l=0.5		11	1	4	0.9167	0.7333	0.8148
	DMSVD	0.5		10	2	4	0.8333	0.7143	0.7692
	HSSCD	d=200, l=0.5		9	3	1	0.7500	0.9000	0.8182
	Proposed SCD	T_1=5, T_2=0.5		10	2	2	0.8333	0.8333	0.8333
HCIL2000 (450)	BMSCD	100000	29	24	5	57	0.8276	0.2963	0.4364
	OFSCD	10		18	11	3	0.6207	0.8571	0.7200
	ATSCD	d=3, l=0.5		21	8	6	0.7241	0.7778	0.7500
	DMSVD	0.5		17	12	1	0.5862	0.9444	0.7234
	HSSCD	d=200, l=0.5		19	10	5	0.6552	0.7917	0.7170
	Proposed SCD	T_1=5, T_2=0.5		26	3	8	0.8966	0.7647	0.8254
Expert Panel And Question Session (450)	BMSCD	100000	7	7	0	6	1.0000	0.5385	0.7000
	OFSCD	10		7	0	0	1.0000	1.0000	1.0000
	ATSCD	d=3, l=0.5		7	0	2	1.0000	0.7778	0.8750
	DMSVD	0.5		6	1	3	0.8571	0.6667	0.7500
	HSSCD	d=200, l=0.5		7	0	4	1.0000	0.6364	0.7778
	Proposed SCD	T_1=5, T_2=0.5		7	0	0	1.0000	1.0000	1.0000
HCIL2006 (500)	BMSCD	100000	20	14	6	24	0.7000	0.3684	0.4828
	OFSCD	10		11	9	22	0.5500	0.3333	0.4151
	ATSCD	d=3, l=0.5		7	13	3	0.3500	0.7000	0.4667
	DMSVD	0.5		8	12	8	0.4000	0.5000	0.4444
	HSSCD	d=200, l=0.5		4	16	0	0.2000	1.0000	0.3333
	Proposed SCD	T_1=5, T_2=0.5		16	4	11	0.8000	0.5926	0.6809
Energy Motion And Proportionality (500)	BMSCD	100000	14	13	1	63	0.9286	0.1711	0.2889
	OFSCD	10		11	3	1	0.7857	0.9167	0.8462
	ATSCD	d=3, l=0.5		13	1	0	0.9286	1.0000	0.9630
	DMSVD	0.5		10	4	2	0.7143	0.8333	0.7692
	HSSCD	d=200, l=0.5		12	2	6	0.8571	0.6667	0.7500
	Proposed SCD	T_1=5, T_2=0.5		12	2	0	0.8571	1.0000	0.9231
Baseball (700)	BMSCD	100000	9	3	6	14	0.3333	0.1765	0.2308
	OFSCD	10		2	7	5	0.2222	0.2857	0.2500
	ATSCD	d=3, l=0.5		6	3	0	0.6667	1.0000	0.8000
	DMSVD	0.5		9	0	0	1.0000	1.0000	1.0000
	HSSCD	d=200, l=0.5		8	1	0	0.8889	1.0000	0.9412
	Proposed SCD	T_1=5, T_2=0.5		9	0	0	1.0000	1.0000	1.0000
Elephant Dream (400)	BMSCD	100000	14	13	1	58	0.9286	0.1831	0.3059
	OFSCD	10		11	3	25	0.7857	0.3056	0.4400
	ATSCD	d=3, l=0.5		12	2	4	0.8571	0.7500	0.8000
	DMSVD	0.5		11	3	1	0.7857	0.9167	0.8462
	HSSCD	d=200, l=0.5		12	2	4	0.8571	0.7500	0.8000
	Proposed SCD	T_1=5, T_2=0.5		12	2	2	0.8571	0.8571	0.8571
Avg.	BMSCD	100000	21.8	18.2	3.6	60.0	0.8222	0.2680	0.3921
	OFSCD	10		15.9	5.9	14.5	0.7018	0.6300	0.6338
	ATSCD	d=3, l=0.5		17.8	4.0	3.9	0.8090	0.8469	0.8133
	DMSVD	0.5		13.3	8.5	2.1	0.7224	0.8520	0.7681
	HSSCD	d=200, l=0.5		14.3	7.5	3.1	0.7374	0.8457	0.7539
	Proposed SCD	T_1=5, T_2=0.5		**19.0**	**2.8**	**4.3**	**0.8923**	**0.8635**	**0.8747**

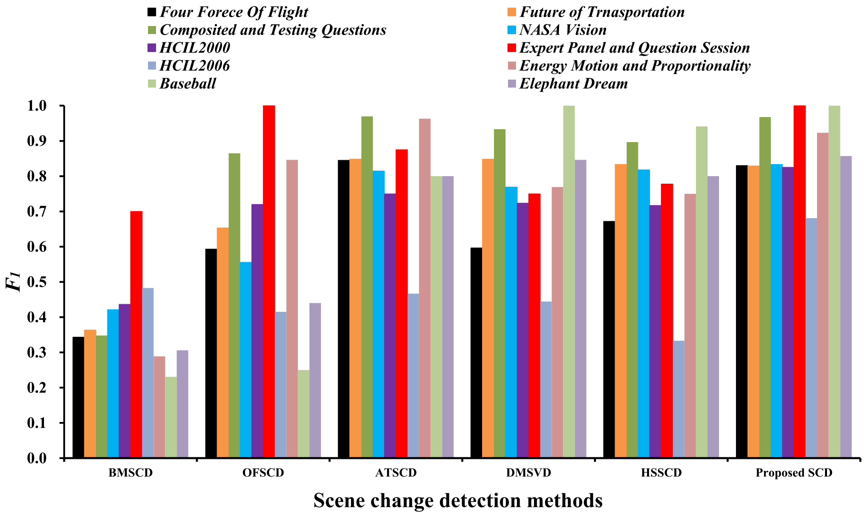
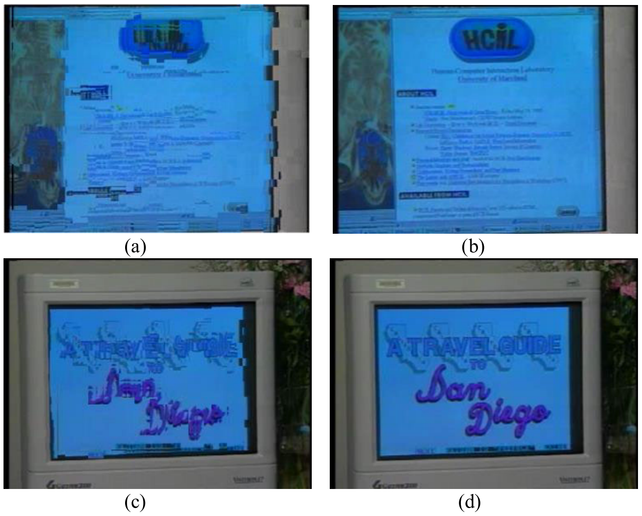


FIGURE 6. Comparison of F1 values for all test sequences.



(a)

(b)

(c)

(d)

FIGURE 7. Comparison of the HCIL2000 interpolated frames generated by: (a) BMSCD, ATSCD, and HSSCD, (b) OFSCD, DMSVD, and the proposed method, (c) OFSCD, DMSVD, and HSSCD, and (d) BMSCD, ATSCD, and the proposed method.

P and *R*. F_1 score range is from 0 to 1, where 1 is the best and 0 is the worst.

Third, we evaluated the computation times of the proposed and benchmark scene-change detection methods to compare the computational complexity. We used five benchmark methods to compare the performance of the proposed method: block matching-bases scene-change detection (BMSCD) [5], optical flow-based scene-change detection (OFSCD) [7], automatic thresholding-based scene-change detection (ATSCD) [9], dual-dissimilarity measure-based statistical video cut detection (DMSVD) [25], and histogram shape-based scene-change detection (HSSCD) [11]. The HSSCD method is the most recent scene-change detection method.

For a fair performance comparison, we used the parameter values guided by their corresponding papers to obtain optimal performance. For the test sequences, we used *Four Forces of Flight* (600 frames), *Future of Transportation* (250 frames), *Composited and Testing Questions* (500 frames), *NASA Vision* (450 frames), *HCIL2000* (450 frames), *Expert Panel and Question Session* (450 frames), *HCIL2006* (500 frames), *Energy Motion and Proportionality* (500 frames), *Base-Ball* (700 frames), and *Elephant Dream* (400 frames) sequences [26]. All test sequences contain both global and local scene changes.

In the first experiment, we compared the quality of the interpolated frames by applying the result of the proposed and benchmark methods to the conventional FRUC algorithm [18]. For the case in Fig. 7, BMSCD [5], ATSCD [9], and HSSCD [11] could not detect the global scene change in the given image. Therefore, when FRUC is applied, incorrect motion information is extracted for the corresponding frame, and interpolated frames with severe visual artifacts are generated within the global scene change (Fig. 7 (a)). For the frames with global scene change, the artifacts from FRUC can be eliminated by repeating the previous frame to generate the interpolated frame.

In contrast, OFSCD [7], DMSVD [25], and the proposed method successfully detected the scene change in the corresponding image. If it is determined that the global scene change has occurred, FRUC using these methods generates the interpolated frame by repeating the previous frame. Consequently, interpolated frames can be generated without visual artifacts within the global scene change (Fig. 7 (b)). For other result images (Figs. 7 (c), (d)), the proposed method was able to generate an improved interpolated frame compared to the benchmark methods used in this paper (BMSCD [5], ATSCD [9], and the proposed method detected the scene change for the given image, while OFSCD [7], DMSVD [25], and HSSCD [11] could not).

TABLE 2. Average computation times per pixel of proposed and benchmark methods.

	BMSCD [5]	OFSCD [7]	ATSCD [9]	DMSVD [25]	HSSCD [11]	Proposed method
Computation time (C_T [μs])	15.5340	5.1067	7.0661	0.0073	5.5391	**2.0108**

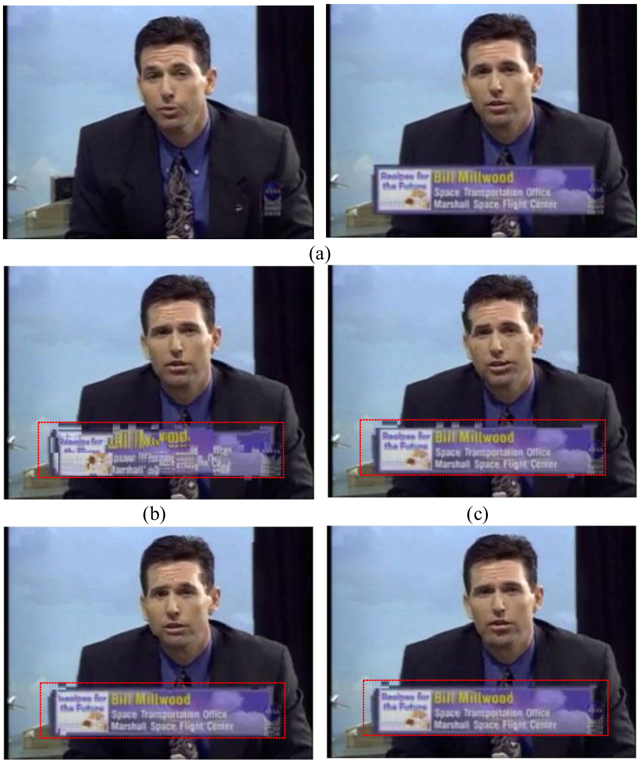


FIGURE 8. Comparison of interpolated frames: (a) two consecutive original video sequence interpolated frames for composite and testing questions test sequences generated by (b) BMSCD (OFSCD and DMSVD produce the same results), (c) ATSCD, (d) HSSCD, and (e) the proposed method.

Furthermore, BMSCD [5], OFSCD [7], and DMSVD [25] could not detect the local scene-change, so the quality of interpolated frames in the region where local scene-change occurs is very low (Fig. 8 (b)). BMSCD [5] and OFSCD [7] detect scene change using motion information, but it is difficult to accurately represent the characteristics of the region only with motion information. Because ATSCD [9] and HSSCD [11] consider the local scene change, their performance is superior to the previous scene-change detection methods [5], [7], [25]. However, since they [9], [11] did not accurately reflect the characteristics of the given local regions, FRUC using these methods also generate the resulting image including some artifacts in the caption boundary regions (Figs. 8 (c), (d)).

In contrast, the proposed method could accurately detect the local scene change, so it improved the image quality of the caption boundary regions during the FRUC process, compared with ATSCD [9] and HSSCD [11] (Fig. 8 (e)). Thus, FRUC with the proposed scene-change detection method could generate a high-quality interpolated frame when the global or local scene change occur.

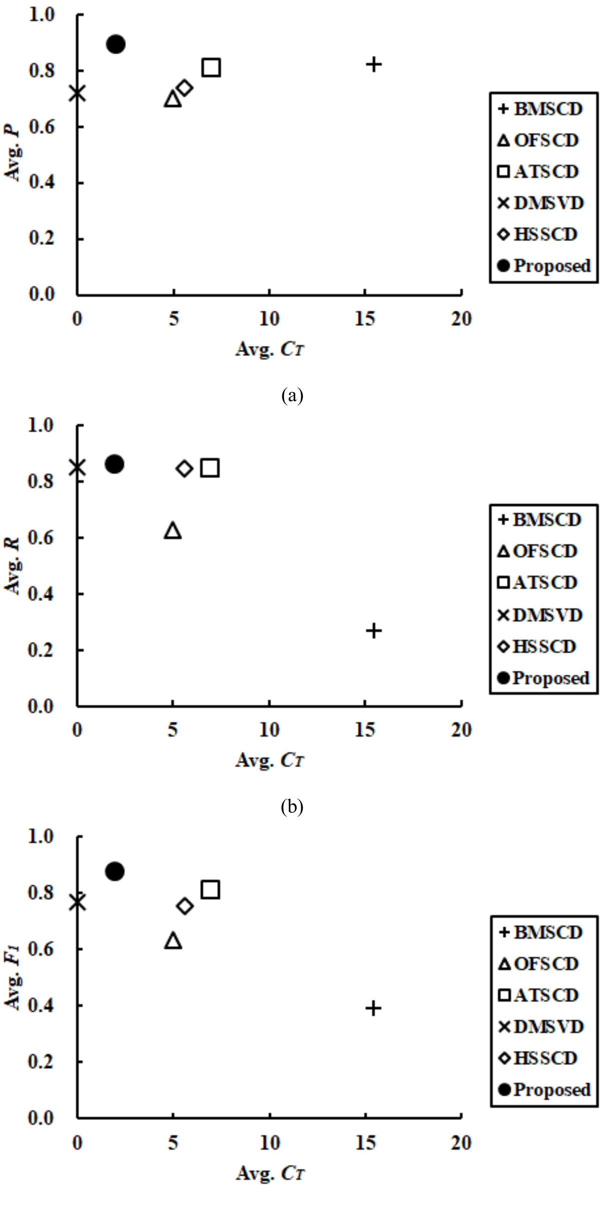


FIGURE 9. Comparison of average (a) P, (b) R, and (c) F1 values and average C_T obtained with the benchmark and proposed methods.

In the second experiment, we compared the accuracy of the proposed and benchmark scene-change detection methods using P, R, and F_1 [9], [11], [22]–[24]. We counted N_{TP}, N_{FP}, and N_{FN} in the video sequences used in our experiments. BMSCD [5] and OFSCD [7] consider only a global scene change without considering a local scene change. Therefore, these methods had lower F_1 scores than the other benchmark methods which consider both local and global scene changes.

DMSVD [25], ATSCD [9], and HSSCD [11] were superior to BMSCD [5] and OFSCD [7] in detecting a scene change and maintaining high F_1 scores. In general, it is more accurate to express the characteristics of the region using luminance information [9], [11], [25] rather than using only motion information [5], [7]. In case of ATSCD [9], they detected the scene change using the difference between the threshold values based on the Otsu method [10] for automatic thresholding. However, in this method, the threshold value obtained from the histogram distribution analysis may not accurately represent the region characteristics of the corresponding frame.

HSSCD [11] used the shape of a 2D histogram to detect a scene change. However, because this method is similar to calculating the luminance difference at the same position of the previous and current frames, the result of this method is the same as the pixel-wise luminance difference. The DMSVD [25] method only considers a global scene change, so it is unsuitable for general FRUC applications. In contrast, the proposed method improved the accuracy of the scene change when compared to benchmark methods [9], [11], and [25], considering both local and global scene changes (Table 1 and Fig. 6).

The proposed method increased the F_1 score by 0.4816 (a 122.51% improvement), 0.2413 (a 38.10% improvement), 0.0741 (a 9.26% improvement), 0.1073 (a 13.98% improvement), and 0.1213 (a 16.10% improvement) compared to the BMSCD [5], OFSCD [7], ATSCD [9], DMSVD [25], and HSSCD [11] methods. The improvement was calculated by dividing the difference between F_1 between the proposed and benchmark methods by the original F_1 score for the benchmark methods [(F_1 score by the proposed method– F_1 score by the benchmark method)/F_1 score by the benchmark method]. The proposed method used the same threshold value for all test sequences and provided the highest accuracy of the scene-change detection.

The performance improvement in the proposed method could be attributed to using the statistical characteristics of the histograms for each region in the given image to extract meaningful values for scene-change detection. That is, the proposed method accurately represents the characteristic value of the given region by expressing the average level of the luminance histogram. Moreover, the proposed method analyzed the distribution of the region surrounding the detected regions to refine the initial scene change regions. With these refinement processes, the proposed method could further improve scene-change detection accuracy when compared to the benchmark methods.

In the third experiment, we calculated the processing times of the benchmark and proposed methods using MATLAB on PC with an Intel i-7-9700K CPU 3.60GHz processor. For the processing time evaluation metric, we used the computation time per pixel (C_T [μs]). The proposed method reduced the C_T by 13.5232 μs (a 87.06% reduction), 3.0959 μs (a 60.62% reduction), 5.0553 μs (a 71.54% reduction), and 3.5283 μs (a 63.70% reduction) compared to the BMSCD [5], OFSCD [7], ATSCD [9], and HSSCD [11] methods, respectively (Table 2). DMSVD [25] had the fastest operation speed but could not detect a local scene change. Therefore, this method had a lower F_1 score than that of the proposed method.

Finally, we compared the overall performance of the proposed and benchmark methods by combining the detection accuracy and the execution time of the scene-change detection algorithms. Fig. 9 illustrates the average P, R, and F_1 values and the average C_T of the benchmark and proposed methods. The proposed method has the highest detection accuracy for scene-change detection in terms of P, R, and F_1 and the fastest operation speed compared with all benchmark methods except DMSVD [25].

IV. CONCLUSION

In this paper, we proposed a new scene-change detection method that uses the analysis of the luminance level in the luminance histogram for the frame rate up-conversion (FRUC). The proposed method divided the input frames into several blocks and then generated luminance histograms. It then removed the meaningless pixel in the generated histograms and extracted the luminance level by calculating the average values in the histograms. We detected the initial scene change regions by calculating the difference between luminance levels for each divided region. Furthermore, we improved scene-change detection accuracy by applying the refinement process that corrects the initial scene change regions.

The experimental results revealed that the average F_1 score of the proposed method was up to 0.4816 (a 122.51% improvement) higher than the benchmark methods. The average processing time per pixel of the proposed method was up to 13.5232 μs (an 87.06% reduction) lower than the benchmark methods except for DMSVD [25]. Furthermore, FRUC using the proposed scene-change detection method could accurately detect local and global scene changes, generating higher-quality interpolated images than the benchmark 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. H. 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, 2004, pp. 309–312.

[4] S.-J. Kang, D.-G. Yoo, S.-K. Lee, and Y. H. 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] H. Shu and L.-P. Chau, ''A new scene change feature for video transcoding,'' in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2005, pp. 4582–4585.

[6] J.-R. Ding and J.-F. Yang, ''Adaptive group-of-pictures and scene change detection methods based on existing H.264 advanced video coding information,'' *IET Image Process.*, vol. 2, no. 2, pp. 85–94, Apr. 2008.

[7] J. Lee, S.-J. Kim, and C. S. Lee, ''Effective scene change detection by using statistical analysis of optical flows,'' *Appl. Math. Inf. Sci.*, vol. 6, no. 1, pp. 177–183, Jan. 2012.

[8] J. A. Manico, J. R. Fredlund, and A. C. Loui, ''Method for automatically organizing a digitized hardcopy media collection,'' U.S. Patent 7 855 810, Dec. 21, 2010.

[9] 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.

[10] L. G. Shapiro and G. C. Stockman, *Computer Vision*. Upper Saddle River, NJ, USA: Prentice-Hall, 2001.

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

[12] K. Doi, R. Hamaguchi, S. Iwase, R. Yokota, Y. Matsuo, and K. Sakurada, ''Epipolar-guided deep object matching for scene change detection,'' 2020, *arXiv:2007.15540*.

[13] S. Ren, K. He, R. Girshick, and J. Sun, ''Faster R-CNN: Towards real-time object detection with region proposal networks,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[14] K. He, X. Zhang, S. Ren, and J. Sun, ''Deep residual learning for image recognition,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[15] L. Ru, B. Du, and C. Wu, ''Multi-temporal scene classification and scene change detection with correlation based fusion,'' *IEEE Trans. Image Process.*, vol. 30, pp. 1382–1394, 2021.

[16] M. Yang, L. Jiao, B. Hou, F. Liu, and S. Yang, ''Selective adversarial adaptation-based cross-scene change detection framework in remote sensing images,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 3, pp. 2188–2203, Mar. 2021.

[17] J. Li, P. Tang, Y. Wu, M. Pan, Z. Tang, and G. Hui, ''Scene change detection: Semantic and depth information,'' *Multimedia Tools Appl.*, pp. 1–19, Apr. 2021.

[18] 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.

[19] 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.

[20] 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.

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

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

[23] 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.

[24] H. S. Lee and S. I. Cho, ''Directional coherence-based scrolling-text detection for frame rate up-conversion,'' *IEEE Access*, vol. 8, pp. 182044–182053, 2020.

[25] G. J. Bae, S. I. Cho, S.-J. Kang, and Y. H. Kim, ''Dual-dissimilarity measure-based statistical video cut detection,'' *J. Real-Time Image Process.*, vol. 13, no. 1, pp. 1–11, Jun. 2017.

[26] G. Marchionini and G. Geisler, ''The open video digital library,'' *Digit. Library Mag.*, vol. 8, no. 12, pp. 1082–9873, Dec. 2002.