IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 22, 2019, accepted January 25, 2019, date of publication February 26, 2019, date of current version March 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2898889

Histogram Shape-Based Scene-Change Detection Algorithm

SUNG IN CHO[1], (Member, IEEE), AND SUK-JU KANG[2], (Member, IEEE)
[1]Department of Electronics and Electrical Engineering, Daegu University, Gyeongsan 38453, South Korea
[2]Department of Electrical Engineering, Sogang University, Seoul 121-742, South Korea

Corresponding author: Suk-Ju Kang (sjkang@sogang.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2018R1D1A1B07048421) and by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2018-0-01421) supervised by the IITP(Institute for Information & communications Technology Promotion.

ABSTRACT This paper proposes histogram shape-based scene-change detection to automatically recognize the changing content in a moving image for frame rate up-conversion; in the frame rate up-conversion, it is important to detect both local and global scene changes. In addition, a threshold value should be fixed regardless of the image characteristics that requires low computation. The proposed algorithm simply extracts the shape of a two-dimensional histogram using point-based distance calculation. It also uses block merging and blocks smoothing to further improve the accuracy of the scene change detection. In the experimental results, the proposed algorithm enhanced the average F1 score to 0.607 (a 189.63% improvement) as compared with the benchmark methods. The average computation time of the proposed algorithm also decreased to 1.696 ms (a 65.81% reduction) compared with the benchmark algorithms.

INDEX TERMS Scene change detection, frame rate up-conversion, and video coding.

I. INTRODUCTION

Scene change detection is a technique for identifying changes in a video sequence, and it has been used for various applications: video indexing, video coding, and frame rate up-conversion. In video indexing, it is important that the relevant information be extracted from the large video database without human intervention [1]. For this, scene change detection automatically recognizes the changes in consecutive frames. In video coding, coding efficiency improves if multiple coding processes are used such as inter-frame and intra-frame predictions [2]; scene change detection is used to enhance effective coding efficiency [3]. In frame rate up-conversion, the high image quality of an interpolation frame is important. Generally, this frame rate up-conversion consists of motion estimation (ME), which calculates a motion vector between consecutive frames, and motion-compensated interpolation (MCI), which produces a new interpolated frame based on the motion vector [4], [5]. If the content of a frame changes locally, ME incorrectly estimates the motion vector, and the image quality of an interpolated frame generated by MCI is degraded. Hence, the scene change detection algorithm for this application should consider both local (i.e., a portion of region in a frame) and global (the whole frame) scene changes.

Various scene change algorithms have been proposed [6]–[11], and these algorithms can be largely classified as motion vector based or histogram based. In motion vector-based algorithms, a block matching-based scene change detection algorithm [6] calculates the differences between previous and current motion vectors after extracting block-based motion vectors from ME. In this case, the accuracy of the motion vector depends on the performance of the scene change detection algorithm. In order to enhance the accuracy, transformed difference-based scene change detection [2] was proposed. This process uses the sum of the absolute transformed difference to calculate the motion vector instead of the sum of absolute difference. An optical flow-based scene change detection algorithm [7] was also proposed. This algorithm also computes the differences between previous and current motion vectors.

For histogram-based algorithms, a scene change detection algorithm using histogram difference [8], [9] was proposed. This generates color histograms and computes the differences between previous and current histograms. Histogram intersection-based scene change detection [10] uses a new

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

matching method and a large database of models to improve the detection accuracy. Chi-square-based scene change detection [11] calculates χ^2 of color histograms in successive frames. Then, it calculates the differences between previous and current χ^2 values.

However, the conventional algorithms have two problems. First, they only consider global scene changes during the process and thus are difficult to use in the frame rate up-conversion. Second, these algorithms cannot use fixed thresholds for the differences between motion vectors or histograms. This is because the threshold should be changed if the image characteristics of the image sequences are different. Multiple histogram-based scene-change detection [3] can solve these problems. It produces histograms for segmented blocks of a frame and automatically computes the optimal threshold. However, it requires considerable computational resources because the automatic thresholding is an iterative algorithm.

In this paper, a histogram shape-based scene-change detection algorithm is proposed. The proposed algorithm is based on a previous study [12], but several operation blocks have been added to improve the performance. Specifically, it generates two-dimensional (2D) histograms for previous and current frames and extracts histogram shape. In addition, it uses block merging and block smoothing to further improve the detection accuracy. Therefore, the proposed algorithm can fix the threshold independent of the image characteristics while requiring low computational complexity.

This paper is organized as follows. Section II explains the proposed algorithm. The experimental results in section III reflect the performance of the proposed algorithm. Finally, section IV concludes the paper.

II. PROPOSED METHOD

The proposed algorithm uses the shape of a 2D histogram that describes the gray-level relationship at the same position for two images. In this histogram, the x axis and y axis denote intensities of the first image and the second image, respectively. Fig. 1 shows the examples of 2D histogram generation; the first and second images are nearly the same, and the distribution of the histogram concentrates on a diagonal ($y = x$ line). If the two images are exactly the same, the histogram will be exactly generated on the diagonal; in contrast, if the two images are different as shown in Figs. 1 (b) and (c), the histogram is randomly distributed as shown in Fig. 1 (e). The proposed algorithm uses this property of the difference for the 2D histogram. Fig. 2 shows the overall block diagram of the proposed algorithm; its specific operation is described in detail as follows.

A. PRE-PROCESSING MODULE

First, the RGB components of the previous and current frames are converted into YCbCr components, and the proposed algorithm uses only Y, which is the luminance. It is defined as follows:

$$P_Y = 0.299 \times P_R + 0.587 \times P_G + 0.114 \times P_B, \quad (1)$$

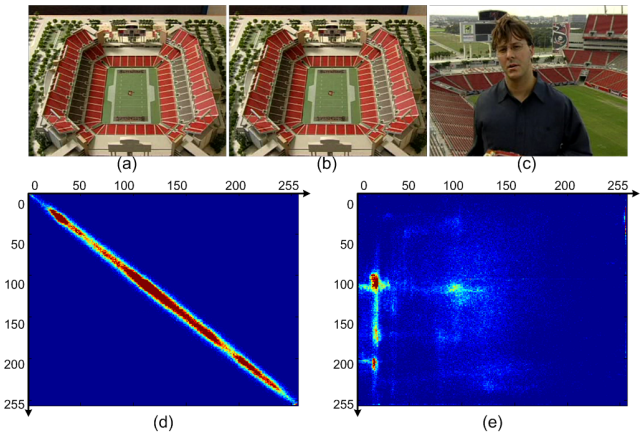

FIGURE 1. Examples of the two-dimensional histogram generation: (a) 1st image, (b) 2nd image, (c) 3rd image, (d) two-dimensional histogram between 1st and 2nd images, and (e) two-dimensional histogram between 2nd and 3rd images.

where P_R, P_G, P_B, and P_Y denote red, green, blue, and Y components, respectively.

Second, in the block partitioning, the previous and current frames with the Y component are divided into several blocks. Then, the block-based 2D histogram processing is performed for each previous and current block.

B. MAIN PROCESSING MODULE

Fig. 3 shows the specific process of the block-based 2D histogram processing. First, the 2D histograms of the previous and current blocks are generated. Second, in the histogram shape extraction, the positions of pixels that are higher than the predefined threshold are extracted to consider only the main pixels in the histogram. Third, the distance is calculated. In the proposed method, a scene change is determined through the shape of the data distribution through a 2D histogram. Fig. 4 shows the principle of the distance calculation. Consider the simplest case when the images of two blocks are exactly the same; in this case, pixels in the histogram exist on the diagonal line. The perpendicular line is drawn from a target point to the diagonal, and the intersection point between the perpendicular line and diagonal is extracted. The distance between the intersection point and the target point (line A) is same as line B and line C as shown in Fig. 4. Therefore, it is calculated as follows:

$$\begin{aligned} D_{i,j} &= \frac{1}{2}\sqrt{(i-j)^2 + (i-j)^2} \\ &= \frac{\sqrt{2}}{2}(i-j), \end{aligned} \quad (2)$$

where i and j denote the coordinate values of the horizontal and vertical axes in the 2D histogram, respectively. $D_{i,j}$ denotes the distance between the intersection point (blue point) and the target point (red point).

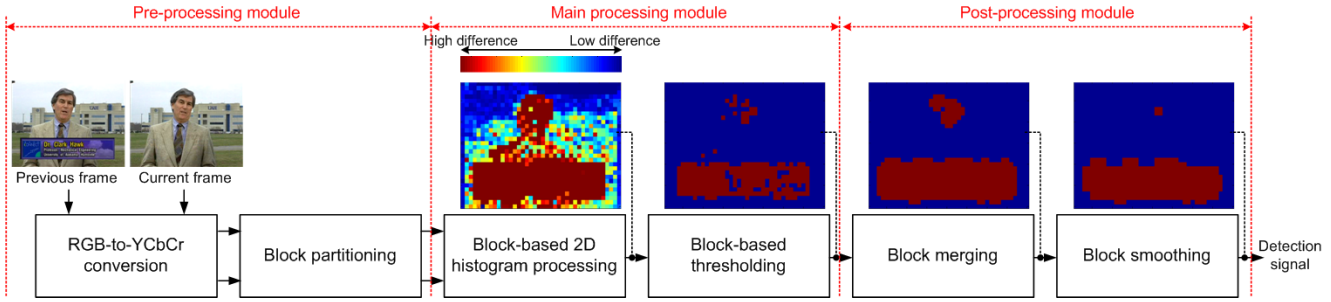

FIGURE 2. Overall block diagram of the histogram shape-based scene-change detection and result images processed by each block.

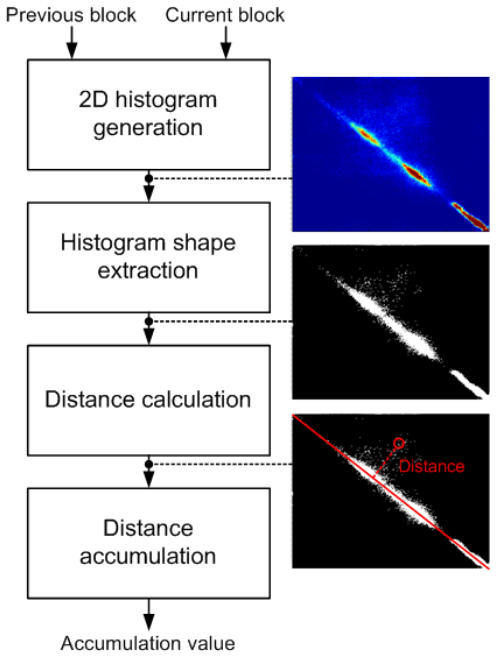

FIGURE 3. Block-based 2D histogram processing and result images processed by each block.

After the distance calculation, the distances of all points in the 2D histogram are accumulated as follows:

$$D^S_{x,y} = \sum_{i=1}^{G_{MAX}} \sum_{j=1}^{G_{MAX}} D_{i,j}, \tag{3}$$

where G_{MAX} denotes the maximum gray level, which is 255 when the pixel depth is 8 bit. x and y denote the horizontal and vertical indexes of a block in a frame, respectively. $D^S_{x,y}$ denotes the sum of distances for a (x, y) block in the 2D histogram.

Then, the block-based thresholding is performed to select blocks that have a greater distance than the predefined threshold because the scene change may occur in these blocks. It is calculated as follows:

$$B_{x,y} = \begin{cases} 1, & if\ D^S_{x,y} > \lambda, \\ 0, & otherwise, \end{cases} \tag{4}$$

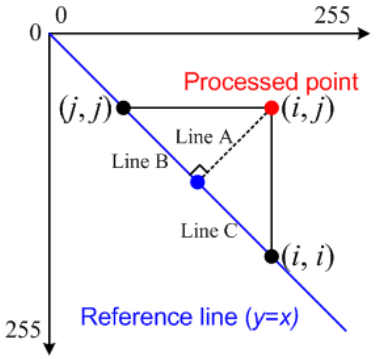

FIGURE 4. Calculating the distance between a processed point and the point of the reference line ($y = x$).

where $B_{x,y}$ denotes a scene-change signal for a (x, y) block. λ denotes a threshold, which is 200. This is based on extensive experiments using test sets.

C. POST-PROCESSING MODULE

In order to improve the detection accuracy of the scene change, the post-processing is performed as in Fig. 2. It consists of two modules: block merging and block smoothing. First, the block merging combines four sub-blocks into one block based on the number of sub-blocks with the scene change as shown in Figs. 5 (b) and (5).

$$CB_{x,y} = \begin{cases} 1, & if\ \left(B_{x,y}+B_{x+1,y}+B_{x,y+1}+B_{x+1,y+1}\right)/4 > \delta, \\ 0, & otherwise, \end{cases} \tag{5}$$

where $CB_{x,y}$ denotes the scene change signals for the combined block considering the number of sub-blocks with the scene change and $CB_{x,y}$ is 1 if the majority of sub-blocks have the scene change ($\delta = 0.5$). This is for robust operation considering problems such as noise and false detection. Therefore, it is defined that a scene change occurs when λ is larger than 0.5, which means more than two of the total of four blocks using the majority rule. In addition, the proposed algorithm considers the additional blocks that overlap with two adjacent original blocks by half (Fig. 5 (b)), while conventional algorithms cannot perform this function (Fig. 5 (a)); therefore, the proposed algorithm enhanced the accuracy of

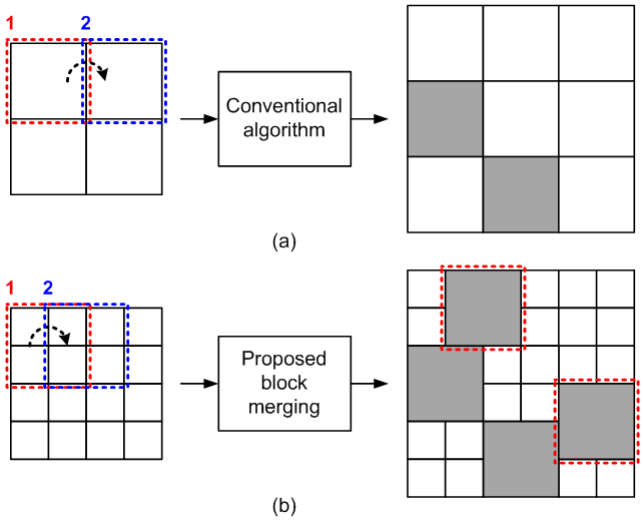

FIGURE 5. Four portions separated by directional distribution analysis: (a) when the state of the current block is corrected into a block with a scene change, (b) when the state of the current block is corrected into a block without a scene change.

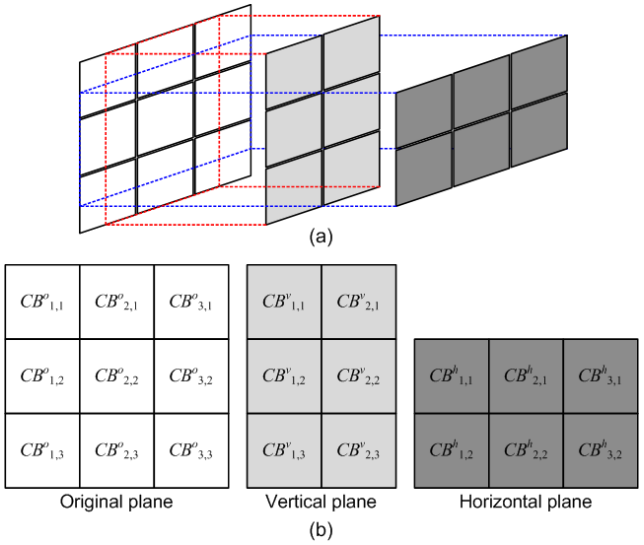

FIGURE 6. (a) Concept of considering overlapped blocks in the original, vertical, and horizontal planes, (b) indexes of blocks in each plane when the number of the considered neighboring blocks is 3.

the scene change detection because it can consider the boundaries between blocks. Next, the block smoothing eliminates the false detection from using the spatial tendency of blocks in a total frame as in multiple histogram-based scene-change detection [3]. However, the proposed algorithm should consider overlapped blocks of horizontal and vertical directions, unlike the conventional algorithm, as shown in Fig. 6. If the majority of the neighboring blocks have the scene change, the final decision of a given block is the scene change as follows:

$$
\begin{aligned}
S^{o}_{x,y} &= \sum_{x=1}^{N}\sum_{y=1}^{N} CB^{o}_{x,y},\\
S^{h}_{x,y} &= \sum_{x=1}^{N}\sum_{y=1}^{N-1} CB^{h}_{x,y},\\
S^{v}_{x,y} &= \sum_{x=1}^{N-1}\sum_{y=1}^{N} CB^{v}_{x,y},\\
CB^{f}_{x,y} &= \begin{cases} 1, & if\ \dfrac{1}{N^2+2N\,(N-1)}\left(S^{o}_{x,y}+S^{v}_{x,y}+S^{h}_{x,y}\right) > \delta,\\ 0, & otherwise, \end{cases}
\end{aligned}
\tag{6}
$$

where $CB^{o}_{x,y}$, $CB^{h}_{x,y}$ and $CB^{v}_{x,y}$, denote the scene-change signals for the block in the original plane and the overlapped blocks in the horizontal and vertical planes, respectively. $S^{o}_{x,y}$, $S^{h}_{x,y}$, and $S^{v}_{x,y}$ denote the sums of $CB_{x,y}$s in the original, horizontal and vertical planes, respectively. N denotes the horizontal (or vertical) number of the considered neighboring block. $CB^{f}_{x,y}$ denotes the final scene-change signal for the combined block, and δ is also 0.5. Using this post-processing module, the proposed algorithm improves the accuracy of the scene change detection further.

III. EXPERIMENTAL RESULTS

In experiments, the proposed and benchmark algorithms were used for the frame rate up-conversion to evaluate the performance. To compare the scene-change detection accuracy, the most popular metrics, which are precision, recall, and F_1 score (F_1) [3], were used as follows:

$$
\begin{aligned}
\text{Precision} &= \frac{NTP}{NTP+NFP},\\
\text{Recall} &= \frac{NTP}{NTP+NFN},\\
\text{F}_1\text{ score} &= 2\times\frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}},
\end{aligned}
\tag{7}
$$

where NFP, NTP, and NFN denote the numbers of false positives, true positives, and false negatives, respectively, and F_1 is the weighted value between 0 to 1; if F_1 is 1, it is the best. Additionally, the interpolated frames were visually compared that were produced by the frame rate up-conversion integrated with either the proposed or the benchmark algorithm. Finally, the computation times of the proposed and benchmarks algorithms were compared; in the algorithms, the block size and the search range size around a given block were 8 pixels × 8 pixels and 16 pixels, respectively. For benchmark algorithms, optical flow-based scene-change detection (OFSCD) [8], block matching-based scene-change detection (BMSCD) [9], automatic thresholding-based scene-change detection (ATSCD) [10], and dual-dissimilarity measure-based statistical video cut detection (DMSVD) [13] were used. For the test sequences, *Energy Motion and Proportionality* (500 frames), *hci2006* (1000 frames), *Expert Panel And Question Session*(1000 frames), and *HCIL2000* (1000 frames) [14] were used, and all test sequences included frames with global or local scene changes.

TABLE 1. Number of false positives.

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
Energy Motion and Proportionality (500)	OFSCD	Low(5)	21	20	1	84	0.952	0.192	0.320
		High(8)		11	10	1	0.524	0.917	0.667
	BMSCD	Low(140)		16	5	29	0.762	0.356	0.485
		High(150)		14	7	17	0.667	0.452	0.538
	ATSCD	d=3, l=0.5		15	6	1	0.714	0.938	0.811
	DDSCD	0.5		11	10	2	0.524	0.846	0.647
	Proposed SCD	=200, =0.5		19	2	1	0.905	0.950	0.927
hcil2006 (1000)	OFSCD	Low(6)	8	8	0	3	1.000	0.727	0.842
		High(10)		6	2	0	0.750	1.000	0.857
	BMSCD	Low(70)		8	0	22	1.000	0.267	0.421
		High(80)		6	2	15	0.750	0.286	0.414
	ATSCD	d=3, l=0.5		7	1	0	0.875	1.000	0.933
	DDSCD	0.5		8	0	2	1.000	0.800	0.889
	Proposed SCD	=200, =0.5		7	1	0	0.875	1.000	0.933
ExpertPanelAnd QuestionSession (1000)	OFSCD	Low(10)	14	7	7	2	0.500	0.778	0.609
		High(14)		7	7	0	0.500	1.000	0.667
	BMSCD	Low(80)		13	1	15	0.929	0.464	0.619
		High(90)		13	1	13	0.929	0.500	0.650
	ATSCD	d=3, l=0.5		14	0	0	1.000	1.000	1.000
	DDSCD	0.5		12	2	1	0.857	0.923	0.889
	Proposed SCD	=200, =0.5		14	0	0	1.000	1.000	1.000
hcil2000 (1000)	OFSCD	Low(6)	12	9	3	13	0.750	0.409	0.529
		High(7)		8	4	9	0.667	0.471	0.552
	BMSCD	Low(90)		11	1	45	0.917	0.196	0.324
		High(100)		9	3	34	0.750	0.209	0.327
	ATSCD	d=3, l=0.5		10	2	3	0.833	0.769	0.800
	DDSCD	0.5		10	2	8	0.833	0.556	0.667
	Proposed SCD	=200, =0.5		9	3	0	0.750	1.000	0.857

First, the detection accuracy of the proposed and benchmark algorithms was evaluated by using the precision, recall, and the F_1 score. Specifically, the number of false positives, the number of true positives, and the number of false negatives for the entire frames for an individual video sequence were calculated, and then the F_1 score was calculated. Table 1 shows the comparison of the scene-change detection accuracy. OFSCD and BMSCD had two different thresholds because the thresholds depend on the image characteristics; when these two thresholds were optimized for precision (a low threshold), the recall decreased, but when they were optimized for recall (a high threshold), the precision decreased. Therefore, these algorithms had lower F_1 scores than the other algorithms. ATSCD and DMSVD had better precision and recall than did OFSCD and BMSCD. The proposed method further enhanced precision and recall, thereby also increasing F_1 by 0.607 (a 189.63% improvement), 0.520 (a 125.56% improvement), 0.280 (a 43.24% improvement), and 0.116 (a 14.30% improvement) over OFSCD, BMSCD, DMSVD, and ATSCD. The improvement was calculated by dividing the increment of F1 (F1 score by the proposed method – F1 score by the benchmark method) by the original F1 for the benchmark method.

Second, the image quality of the interpolated frames for the proposed and benchmark methods was visually compared. Fig. 7 (a) shows consecutive original frames that have a local scene change, of the *Energy Motion and Proportionality*test sequence, and Figs. 7 (b) – (e) show interpolated frames. OFSCD, BMSCD, and DMSVD could not consider a local scene change (it was assumed that block matching is used as a default motion estimation technique in DMSVD). Hence, motion estimation wrongly estimated motion vectors, and interpolated frames were degraded by artifacts. In case of ATSCD, the accuracy of the scene-change detection was higher than that with OFSCD and BMSCD. However, it also had some artifacts in the region with local scene change as shown in Fig. 7 (d). In contrast, the proposed algorithm enhanced the detection accuracy, and the image quality of the interpolated frame was also significantly improved.

Third, the average computation times of the proposed and benchmark algorithms for each test sequence were compared; Table 2 shows the average times. The proposed algorithm

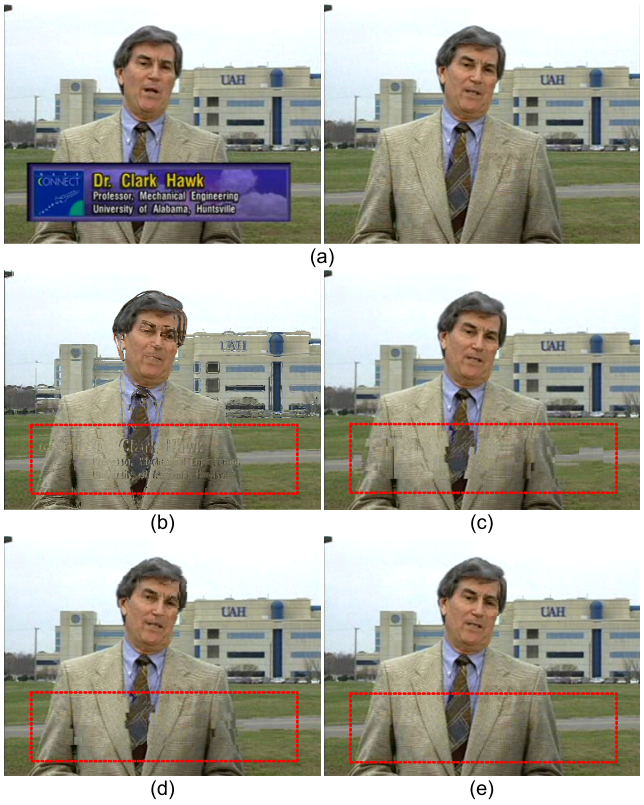

FIGURE 7. (a) Consecutive original frames and interpolated frames of the energy motion and proportionality test sequence generated by (b) OFSCD, (c) BMSCD (or DMSVD), (d) ATSCD, and (e) the proposed SCD.

TABLE 2. Average computation time of the proposed and benchmark methods.

OFSCD	BMSCD	ATSCD	DMSVD	Ours
2.459	2.577	2.212	0.103	0.881

greatly reduced the average computation time, by 1.578 ms (a 64.17% reduction), 1.696 ms (a 65.81% reduction), and 1.331 ms (a 60.16% reduction), respectively, compared to OFSCD, BMSCD, and ATSCD. DMSVD greatly decreased the computation time, but it could not be used in the local scene-change detection.

IV. CONCLUSION

In this paper, a new scene-change detection algorithm based on a histogram shape extraction was proposed for the frame rate up-conversion. Specifically, the proposed algorithm calculated the distance for each point after generating a 2D histogram, to extract the histogram shape. Additionally, it combined the sub-blocks into a large block and corrected a detection signal of a current block considering the neighboring blocks to enhance the algorithm performance further. The experimental results show that the average F_1 score of the proposed algorithm was up to 0.354 (a 63.99% improvement) higher than those for the benchmark algorithms. The proposed algorithm also reduced average computation time by up to 1.696 ms (a 65.81% reduction) over the benchmark algorithms.

REFERENCES

[1] R. Brunelli, O. Mich, and C. M. Modena, "A survey on the automatic indexing of video data," *J. Vis. Commun. Image Represent.*, vol. 10, no. 2, pp. 78–112, Jun. 1999.
[2] 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.
[3] 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.
[4] 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.
[5] 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.
[6] 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.
[7] J. Lee, S.-J. Kim, and C. S. Lee, "Effective scene change detection by using statistical analysis of optical flows," *Appl. Math. Info. Sci.*, vol. 6, no. 1, pp. 177–183, Jan. 2012.
[8] J.-R. Kim, S. Suh, and S. Sull, "Fast scene change detection for personal video recorder," *IEEE Trans. Consum. Electron.*, vol. 49, no. 3, pp. 683–688, Aug. 2003.
[9] C. F. Lam and M. C. Lee, "Video segmentation using color difference histogram," in *Lecture Notes in Computer Science*. Springer, Aug. 1998, pp. 159–174.
[10] M. J. Swain and D. H. Ballard, "Color indexing," *Int. J. Comput. Vis.*, vol. 26, no. 4, pp. 11–32, Nov. 1991.
[11] A. Nagasaka and Y. Tanaka, "Automatic video indexing and full-video search for object appearances," in *Proc. IFIP 2nd Work. Conf. Vis. Database Syst.*, 1992, pp. 113–127.
[12] S.-J. Kang, "Positional analysis-based scene-change detection algorithm," in *Proc. IEEE Int. Conf. Consum. Electron.*, Jan. 2015, pp. 11–12.
[13] 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.
[14] G. Marchionini and G. Geisler, "The open video digital library," *Digit. Library Mag.*, vol. 8, no. 12, pp. 1082–9873, Dec. 2002.

SUNG IN CHO (S'10–M'17) 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. He is currently an Assistant Professor of electronic engineering with Daegu University. His current research interests include image analysis and enhancement, video processing, multimedia signal processing, deep learning algorithms, and circuit design for LCD and OLED systems.

SUK-JU KANG (S'08–M'11) received the B.S. degree in electronic engineering from Sogang University, South Korea, in 2006, and the Ph.D. degree in electrical and computer engineering from the Pohang University of Science and Technology, in 2011. From 2011 to 2012, he was a Senior Researcher with LG Display, where he was a Project Leader for resolution enhancement and multi-view 3D system projects. From 2012 to 2015, he was an Assistant Professor of electrical engineering with Dong-A University, Busan. He is currently an Associate Professor of electronic engineering with Sogang University. His current research interests include image analysis and enhancement, video processing, multimedia signal processing, circuit design for display systems, and deep learning systems.