Fluctuation-Based Fade Detection for Local Scene Changes

SANGHO YOON¹, (Student Member, IEEE),
CHANHEE LEE¹, (Graduate Student Member, IEEE),
HO SUB LEE², (Member, IEEE), YOUNG HWAN KIM¹, (Life Senior Member, IEEE),
AND SEOKHYEONG KANG¹, (Member, IEEE)

¹Department of Electrical Engineering, Pohang University of Science and Technology (POSTECH), Pohang 790-784, Republic of Korea
²Department of Electronic Engineering, Daegu University, Gyeongsan 38911, Republic of Korea

Corresponding author: Seokhyeong Kang (shkang@postech.ac.kr)

This work was supported in part by the Ministry of Science and ICT (MSIT), South Korea, under the ICT Creative Consilience Program supervised by the Institute for Information and Communications Technology Planning and Evaluation (IITP), under Grant IITP-2020-2011-1-00783.

ABSTRACT In recent years, fade detection algorithms can classify fade scenes in massive video libraries have been developed. However, these algorithms misclassify some non-fade scenes as fade scenes, especially dissolve scenes and scenes with captions or flashing light sources. This paper proposes a new fade detection algorithm that uses similarity tendencies of luminance transitions to overcome such obstacles. To prevent detection accuracy degradation by letterboxing and captions, video frames are simplified. Then, fade candidates are detected by transition boundary detection using the angular and curvature characteristics of the luminance vectors. Finally, luminance flipping detection improves the detection accuracy by extracting the luminance retrograde phenomenon that occurs with flashing or light source movements. Through objective evaluation using $F_1$ score, the detection accuracy of the proposed algorithm was 0.884, which is an increase of 0.187 (21.2% improvement) compared with the average $F_1$ score of existing high-performance methods.

INDEX TERMS Algorithm design and analysis, change detection algorithms, machine learning algorithms, video signal processing.

I. INTRODUCTION

Over the past 20 years, the demand for video management operations, such as browsing and indexing, has rapidly increased due to the widespread production and consumption of videos. Managing an enormous content-based video database requires a good video categorizing system. The first step of categorizing video content in video management is to classify the boundaries of shots. A shot is defined as a series of frames that capture continuous spatiotemporal action with one camera, which means that the video content changes when the shot changes [1]. The boundary of two continuous shots is defined as a scene change or transition. If scene transitions are clearly detected, a video database can be managed through the classification of video content. In addition, a shot boundary detection algorithm can be utilized as a preprocessing algorithm for frame rate up-conversion [2]–[4] to improve video quality.

Scene changes can be categorized as abrupt or gradual. An abrupt scene change is relatively easy to detect as the shot is completely changed in only a single frame. However, a gradual scene change is difficult to detect accurately because the shots are switched over several frames. The types of gradual scene change can be classified as fade, dissolve, or wipe. Fade is a shot transition technique between one shot and a monochromatic frame. Fade-out is a scene change where a shot transition into a monochromatic frame, and fade-in is the opposite. Dissolve and wipe are gradual scene change techniques between two consecutive shots. Dissolve changes all pixel values at a specific rate over time, and wipe determines the number of pixels to change at a specific rate over time. Among the gradual scene change types, fade is difficult to detect accurately due to the various types of outlier scenes, as demonstrated in Fig. 1.
There are three main scene types that can be easily confused in detecting a fade scene change (Fig. 1). This can be represented by the standard deviation of luminance used in general gradual scene change detection (Fig. 1a). First, fade and dissolve are distinguished by whether the start or end frame is a monochromatic frame. However, if the overall luminance of one or more frames is low, a dissolve may be erroneously detected as a fade. This can be resolved through a clear definition of the monochromatic frame. Second, there are also cases where a dissolve can be incorrectly recognized as a fade due to local light changes. It is typical for there to be a flashing scene in which much light is generated or disappears (Fig. 1c). This is not strictly a scene change, but its characteristics are very similar to those of a fade, making it difficult to detect fades in such cases. Third, there are cases where a luminance transition appears similar to a fade due to the movement of a light source or object (Fig. 1d). Because this shows a luminance transition tendency similar to that of a fade, it is difficult to distinguish it from a fade.

Therefore, in this paper, we propose a new method to improve fade detection accuracy. To detect a fade scene accurately, we propose the following three steps. First, valueless pixels in the frame are removed by simplification. This is to accurately detect monochromatic frames that must exist in a fade scene. In this process, pixels irrelevant to scene changes, such as captions and letterboxing, are ignored. Second, the scene change boundary is detected through vector analysis of luminance transition. To detect a gradual scene change, it is necessary to clarify the boundary between the beginning and end of the scene change. As shown in the graph of luminance standard deviation (Fig. 1a), the scene change boundary is detected by the point at which the luminance vector changes rapidly. Through this, the candidate for the fade section is determined. Finally, the outlier that is not the actual scene change is removed. In particular, this phenomenon is evident in the case of a scene where the light changes locally. Fade generally decreases the entire luminance of the frame at a certain rate by its mathematical definition. Therefore, the luminance tendency tends to remain constant during a fade scene. Meanwhile, a luminance flipping phenomena, such as light reflection, can occur at object boundaries when flashing or light source movement occurs in the scene. By detecting the luminance flipping phenomenon and calculating the luminance tendency similarity, the light source movement or flashing can be excluded to better detect fade.

In summary, this paper makes the following three contributions:

1) We propose a preprocessing process to remove artificial effects such as captions and letterboxes as outlier pixels, since they are irrelevant to scene changes.
2) We propose a luminance transition vector to clearly distinguish fade and dissolve, and accurately detected all intervals where fade occurred without missing frames, which often occur in existing fade detection algorithms.
3) Since the flashing scene has similar fade and luminance changes, it can be falsely detected as a fade and reduce precision. In order to distinguish a flashing scene from a fade, we propose an algorithm to detect luminance flipping, which is a phenomenon in which the luminance fluctuation appears differently between an object boundary and the background.

In our experiments, the proposed method outperformed existing methods on average in terms of $F_1$ score. Specifically, it improved the detection accuracy by 4.1% in...
frame-based evaluation and 5.3% in transition-based evaluation. Moreover, it showed a high detection accuracy of more than 90% for a movie trailer dataset, which is a high-resolution video dataset.

The remainder of this paper is organized as follows. Section 2 describes work related to fade detection. Section 3 presents the mathematical definition of a fade scene. Section 4 describes the proposed fade detection algorithm. Section 5 demonstrates the accuracy of the algorithm compared with existing methods. Finally, Section 6 concludes the paper.

II. RELATED WORKS
Various fade detection methods have been proposed based on the definition of a fade. At the beginning of the research on the fade detection, pixel-based approaches (PBAs) have been proposed to detect a fade based on the phenomenon that the estimated luminance increases or decreases across the frame [5], [6], [29]. Reference [5] estimates the motion of pixels using the block matching algorithm, and detects the fade by using the luminance transition of each pixel. On the other hand, [6] detects fade-in and fade-out by using whether the luminance of each pixel in the frame decreases or increases at a certain rate, respectively. Reference [29] uses earth mover’s distance (EMD) to express the movement of pixel values with precise and meaningful values. EMD is defined as the total amount of work required for each pixel to transition to a different pixel value. Thus, the transition of EMD is smooth, it is defined as a gradual scene change. However, because a PBA is basically suitable for a static scene, it has the disadvantage that it is very vulnerable to fast object motion or camera panning.

For extension to the detection of dynamic scenes, histogram-based approaches (HBAs) have been proposed [7]–[12], [25], [27], [28]. A HBA detects a fade not by using the luminance transition of each pixel but by the transition of the distribution of pixels in the frame. However, to use the pixel distribution, a HBA must assume that a fade maintains a steady state with low diversity of the background in consecutive frames. Under this assumption, HBAs can overcome the limitations of PBAs and detect fades in a dynamic scene. In the [8], [9], gradual scene change is detected by using color histograms to see whether the variation in the distribution of histograms continuously decreased or increased. In the same way, [25] estimates the difference of color histograms as a frame dissimilarity to detect shot boundary. However, color histogram has a disadvantage that the value can change greatly depending on the direction of light or shadow. In order to overcome this problem, a fuzzy color histogram-based method that categorizes color bins largely is proposed in [7] and [10]. Meanwhile, [11] proposes an algorithm based on just noticeable difference (JND), focusing on the fact that frame change is determined by human perception. Reference [12] also detects gradual scene change based on the similar difference in color histogram difference between consecutive frames apart with specific intervals. In addition, [27] and [28] express the change of histogram as mutual information (MI), and detect the scene change, using the difference in the amount of information occurring between successive frames. However, if they have fast motion of objects, histogram-based feature values such as JND and MI fluctuate greatly, thus reducing the detection accuracy.

Edge-based approaches (EBAs), which use the feature that the amount of information changes rapidly when the scene changes, have also been proposed [13]–[18], [26]. An EBA detects the edge, calculates the amount of change in the edge, and compensates for motion to determine how the information of the scene has changed. References [13] and [14] propose a method to detect scene change when the transition of total amount of edge in the consecutive frames is large. However, it is not suitable for gradual scene changes where the edges change slowly or become blurred. To solve this problem, in [15]–[17], scene change is detected using the creation and loss of edge by tracking the edge in the frame. On the other hand, [18] combined edge transition with the change of pixel intensity to detect gradual scene changes. However, it is more suitable to detect dissolve than fade, and difficult to distinguish between fade and dissolve. Reference [26] defines the effective average gradient (EAG), and detects the shot boundary using the property that the total amount of edges in each frame remains constant until the shot is changed. However, there is a disadvantage in that it is difficult to detect a gradual scene change for a flashing scene or a low-intensity scene where edges are easily lost.

One other method [19] detects the boundary of a shot using a self-similarity matrix. Because the self-similarity matrix represents the degree of similarity between two consecutive frames in the frequency domain, abrupt scene changes with a large information transition can be accurately detected. However, detection for a gradual scene change is difficult in the case of a low-intensity frame with little change in information or a fade that progresses for a long time.

Recently, statistical approaches (SAs) have been proposed [20], [21]. An SA generates global and local features from the frame, and models a mathematically appropriate function to classify the shot using these features. An SA adopts probabilistic classification to distinguish extracted features. However, these approaches are greatly dependent on whether the features can represent the shot. To avoid statistical bias in an SA, the training data should represent the shot’s characteristics.

Recently, scene change detection algorithms based on deep neural networks have been proposed [31]–[33]. Reference [31] performs spatial and temporal analysis using deep 3D-CNN, and applied it to a single vector machine (SVM) to detect transition types. However, there is a false detection by little motion, which requires post-processing to correct it. [32] implements the binary problem of determining whether the current frame is the same shot as the previous frame using CNN. However, gradual scene change has a frame duration, so it has to be specified. The network in [33] is composed of subnets that extract features of each frame using
a deep 3D-CNN-based feature extractor and categorize shots through this similarity. It is characterized by estimating the transition length of gradual scene change through multi-scale windowing.

Based on research on gradual scene change methods, our proposed fade detection method uses the common characteristics of a scene change and differential characteristics from other scene change techniques, such as dissolve. To detect a fade, we use the specific luminance distribution in terms of mathematically defined scene changes.

Our method is a histogram-based approach that can detect fades in real scenes as well as in mathematically defined fades. The proposed method can accurately find transition boundaries using luminance vectors, and it improves the detection accuracy by using the feature that the temporal luminance transition in a dissolve and the spatial luminance transition in a flashing scene appear differently from those in a fade.

### III. BACKGROUND
#### A. SCENE CHANGE
A scene change can be categorized by two types: abrupt and gradual scene change. An abrupt scene change is a transition to which certain effects are not applied. Existing work on shot boundary detection has focused on abrupt scene change detection because of its simple structure [15]. Early work on abrupt scene change detection typically employed a threshold of feature difference between frames [22]. A gradual scene change is a shot transition spanning several frames, characterized by a gradual change in luminance or pixel values. It is generated by the post-production process of video editing. Even a gradual scene change has unclear boundaries and must be clearly detected for the accurate management of video databases. For this reason, previous works have focused on detecting fade effects, which are a representative type of gradual scene change. By analyzing these scene changes in real video data, their properties can be classified by the standard deviation (STD) of the luminance of each frame (Fig. 1a). In the case of an abrupt scene change (i.e., CUT), the STD of the luminance changes significantly when the shot changes. Meanwhile, in a gradual scene change, the value changes relatively smoothly as the shot changes.

#### B. MATHEMATICAL DEFINITION OF FADE
A fade is a transition characterized by a gradual decrease (fade-out) or increase (fade-in) in visual intensity. Unlike abrupt scene change detection, gradual scene change detection has been attempted to accurately find the starting and ending frames. In particular, research on video management has mainly involved the detection of dissolve and fade effects. Fade-out is a video transition that darkens the frames until the frame is almost a black screen. By contrast, fade-in lightens the frame from a blank frame to a fully illuminated frame. Expanding on its definition, a fade can transition to/from any monochromatic frame, not just a black frame. According to this expanded definition, a fade scene can be defined as a process that fills the shot boundary with a continuous video frame between one shot and a monochromatic frame. It can be mathematically modeled with luminance scaling. Let $G(x, y, t)$ be the sequence in grayscale, $P(x, y)$ be the start frame, $Q(x, y)$ the end frame, $t$ the frame number, and $l_t$ the length of transition sequence. An intensity-scaled fade scene is modeled as follows:

$$F(x, y, t) = G(x, y, t) \cdot (1 - \frac{t}{l_t}), \quad t \in [t_0, t_0 + l_t].$$  \tag{1}

Fade-out is modeled as follows:

$$F(x, y, t) = P(x, y) \cdot (\frac{l_t - t}{l_t}).$$  \tag{2}

Fade-in is modeled as follows:

$$F(x, y, t) = Q(x, y) \cdot (\frac{t}{l_t}).$$  \tag{3}

The above formulas are primarily applied to static scenes. They are not generally applied in practical situations. Therefore, the proposed algorithm uses the following lemma.

#### C. LEMMA FROM DEFINITION OF FADE
In ideal cases, a gradual scene change can be expressed as a mathematical model. Each pixel of the start and end frames can be defined using $\alpha$ and $\beta$, which are functions with respect to time $t$. As fade and dissolve have less spatial dependence, $\alpha$ and $\beta$ can be defined as follows:

$$F(x, y, t) = \alpha(t) \cdot P(x, y) + \beta(t) \cdot Q(x, y).$$  \tag{4}

For a fade-in, the start frame $P(x, y)$ becomes a monochromatic frame. For a fade-out, the end frame $Q(x, y)$ becomes a monochromatic frame. In the case of fade and dissolve, $\alpha$ is a monotonically decreasing function, and $\beta$ is a monotonically increasing function. The relationship between $\alpha$ and $\beta$ satisfies the following equation:

$$\alpha(t) = 1 - \beta(t).$$  \tag{5}

For application to a more practical situation, the proposed algorithm employs detection using the distribution instead of the fade definition applied to each pixel. This helps in detecting fades in dynamic scenes. The details are described in transition boundary detection.

### IV. LUMINANCE VECTOR-BASED FADE DETECTION ALGORITHM
A fade detection algorithm classifies a fade scene from other scene changes. Fade is a scene change technique in which the luminance continuously decreases or increases over several frames. Therefore, a fade scene is generally detected based on the luminance transition, and the process of a typical fade detection algorithm is shown in Fig. 2. First, it is essential to remove various outlier pixels, such as captions, before detecting fade scenes. Therefore, fade detection algorithms start by simplifying the frames. As a result of simplification, frames are converted into grayscale. After simplification, transition sections where the luminance continuously
increases or decreases are found in the overall video frames. After finding all transition sections, fades are classified in the video sequence.

In this paper, we propose a new fade detection algorithm that consists of three main steps (Fig. 3). First, frame simplification is performed to improve the detection accuracy regardless of the image quality and resolution. As a fade is a gradual scene change that monotonically increases or decreases luminance in the frames, all frames are converted to grayscale. In addition, monochromatic frames are detected by the gray level of the frame. After simplification of all video frames, all transitions that could be gradual scene changes are detected at the frame level. Following the luminance transition tendency of all frames, fade candidates are found and transition boundaries are clarified. To detect the transition boundary accurately, a luminance transition graph is generated. Assuming the gradient of the graph as the luminance vector, the transition boundaries that can be detected as fade scenes are detected. After detecting all fade-like transitions, outliers such as dissolve, flashing, and light source movements are classified using luminance flipping detection. After classification, the fade detection algorithm distinguishes a fade from other scene changes.

The following subsections describe each of the steps of the proposed algorithm: frame simplification, transition boundary detection, and luminance flipping detection as illustrated in Fig. 3.

A. FRAME SIMPLIFICATION

Preliminary steps are performed to ensure that fades can be detected accurately, even if letterboxing and captions are included. First, letterboxing is removed from the edge of the frame. A letterbox is an artificially edited area that helps fit the aspect ratio of the frame to the display. Removing letterboxing means erasing the single-color area around the frame boundary. To standardize the quality and resolution, video frames are then downsampled to \(320 \times 240\) resolution. With this process, the luminance of the frame increases or decreases for all the pixels simultaneously. A fade will not be affected in terms of the detail of the image. Therefore, this process reduces the amount of information in the frame without compromising detection performance. Finally, the proposed method converts the RGB color space into grayscale to detect a fade scene defined by the luminance transition. The caption outlier removal process reduces the chances of a lower detection accuracy. As a result of the outlier pixel removal process, frames are standardized with the same resolution and converted to grayscale to detect fades based on their definition.

B. TRANSITION BOUNDARY DETECTION

This process finds candidates that are detected as fades. The detected candidates contain the frame numbers corresponding to when the fade starts and ends as detected luminance vector-based detection. Existing fade detection methods usually classify the shot by the STD of the luminance value of a frame. This focuses on the fact that the luminance of a monochromatic frame converges to one value with a small variance.

The proposed method does not use the STD of the luminance from all pixels. Instead, it uses all the pixels except the edge pixels. Videos generally have artificially edited...
effects in addition to shots incorporating natural conditions. A static caption is an artificial edit effect. Because captions are composed of text or figures and occupy the textured part, edge pixel removal can exclude such textured areas. In our experiment, Canny edge detection [23] was used to extract the details of all edges, making it possible to obtain the luminance distribution for the background pixels without the edge pixels, which increases the variance of the luminance, as shown in Fig. 4.

After calculating the STD of the luminance for the remaining pixels excluding the edge, a monochromatic frame can be detected as falling within a very small STD (0.03 or less in our experiment). Around all of the detected monochromatic frames, fade and dissolve scenes are distinguished by the characteristics of the luminance transition vector.

Detecting the monochromatic frames can determine the time intervals of the fade candidates. The luminance transition vector is calculated as the difference in the luminance STDs among adjacent frames. The start and end frames of a fade candidate are determined by thresholding the angle with $\theta_{th}$ between the average change and instantaneous change in the luminance STD (Fig. 5a).

After determining the length of a fade candidate, the proposed method determines whether the detected candidate is a fade or dissolve outlier. As mentioned in Section 3, $\alpha$ continuously decreases, while $\beta$ continuously increases during the gradual scene change. However, because they do not change ideally in actual video frames, the distribution of the luminance appears to follow this tendency. Therefore, it can be described in terms of the variance of the start and end frames as follows:

$$\sigma^2(F) = \alpha^2(t) \cdot \sigma^2(P) + \beta^2(t) \cdot \sigma^2(Q).$$  \(6\)

A fade effect decreases the luminance STD value to zero at some point in a scene, where the start or the end frame is completely composed of one color (monochromatic frame). Unlike a fade effect, a dissolve effect usually does not converge to zero STD, even though the scene looks like a monochromatic frame. This phenomenon is described as a graph in Fig. 5b. In this graph, the left curvature is a dissolve effect, and the right curvature is a fade effect. The dissolve has a more arc-like shape than the fade, which could be described by a quadratic equation. The curvature calculation of these two effects can be expressed as follows:

$$k(x) = \frac{2a}{(1 + (2ax + b)^2)^{3/2}}.$$  \(7\)

Through thresholding, if the curvature is sufficiently large, the measured curvature with $k_{th}$ approaches the shape of a small circle, which means that the graph has a parabolic shape. In this situation, the scene is classified as a dissolve scene. Otherwise, when the curvature is small, the graph approaches a straight line. In this case, the scene is classified as a fade (Fig. 5b).

C. LUMINANCE FLIPPING DETECTION

Finally, the proposed algorithm distinguishes fades from the incorrectly classified candidates, such as light source. There are some cases in which the movement of the light source in a scene behaves like a fade effect. The major difference between a fade effect and flashing outlier is a steady change in the luminance. In a fade effect, the luminance of the entire frame gradually changes to or from a monochromatic frame. For the cases in which the light source is moving, a steady luminance change does not occur. When a foreground object and a background exist, the light and shadows are distinguished, as shown in Fig. 6b. The reflected light and shadows
cast on the boundary move along the boundary. As a result, the pixels that have different transition directions from the overall tendency are gathered around the object boundary. For example, the darkness caused by the movement of the light source also has a pixel whose luminance increases as shown in Fig. 6b. However, in the case of fade-out, the luminance of all pixels decreases as shown in Fig. 6c. To detect this phenomenon, the proposed method generates a dilated edge region by expanding the boundary region detected by Canny edge detection. In the proposed method, the dilated edge pixels are written as the boundary (BD) region, and the entire region is written as the background (BG) region.

To discover a different tendency of the luminance transition, the luminance transition tendency (LTT) is calculated. The LTT is the ratio of the change in boundary luminance that deviates from the change in background luminance, which is as calculated follows:

$$\text{LTT} = \frac{\text{LIT}_{\text{BD}}}{\text{LIT}_{\text{BG}}}.$$  (8)

If LTT is 1, the tendency is considered to be consistent, and it is detected as a fade. If LTT is less than 0.5, the tendency is considered to be inconsistent. In this case, a tendency similarity (TS) is also measured.

The TS is largely composed of the STD of BG, and mean and STD of BD. The difference between the tendency and STD of BD corresponding to the control is calculated. $T_{BG, \text{std}}$ indicates the background STD of the luminance, and $N$ indicates normalization from 0 to 1. To analyze the luminance transition tendency of the light source movement and fade-out, the graphs of normalized luminance transitions are shown in Fig. 6a. The NT values steadily decrease upon fade-out, while the light source shifts show different trends in the NT values. As the importance of the similarity increases in the middle of the transition, the weight of TS is calculated as follows:

$$w_{TS} = NT_{BG, \text{std}} \cdot (1 - NT_{BG, \text{std}}).$$  (9)

The TS of the boundary area and final TS are calculated as follows:

$$TS_{\text{mean}} = \frac{|NT_{BG, \text{std}} - NT_{BD, \text{mean}}| \cdot w_{TS}}{1}.$$  (10)

$$TS_{\text{std}} = \frac{|NT_{BG, \text{std}} - NT_{BD, \text{std}}| \cdot w_{TS}}{1}.$$  (11)

$$TS = \sqrt{TS_{\text{mean}}^2 + TS_{\text{std}}^2}.$$  (12)

As a result of calculating the tendency similarity of the light source movement and fade-out, the graphs of tendency similarities are shown in Fig. 6b. This shows a dramatic variance in light source movement compared to fade-out. If TS is $TS_{th}$ or more, it is determined that the luminance transition is less similar and changes irregularly. In our experiment, $TS_{th}$ was set as 0.9 because a value representing the best performance was estimated. This is excluded from the fade scene to complete the detection algorithm.

V. EXPERIMENTS

We conducted various experiments to evaluate the detection accuracy of the previous fade detection algorithms [26], [27], [29], [31]–[33] and the proposed fade detection algorithm.

| Video | Resolution | Frame count | # of fade transitions | # of fade frames |
|-------|------------|-------------|-----------------------|------------------|
| TREC01 | 352×240 | 112,087 | 35 | 518 |
| TREC02 | 352×240 | 22,659 | 3 | 62 |
| TREC03 | 352×240 | 58,142 | 9 | 176 |
| TREC04 | 352×240 | 58,142 | 22 | 531 |
| TREC05 | 352×240 | 58,142 | 9 | 214 |
| TREC06 | 352×240 | 58,142 | 33 | 803 |
| TREC07 | 352×240 | 12,863 | 2 | 101 |
| TREC08 | 352×240 | 58,142 | 37 | 541 |
| TREC09 | 352×240 | 9,772 | 4 | 79 |
| TREC10 | 352×240 | 28,502 | 6 | 158 |
| Total | | | 476,593 | 160 | 3,183 |
To evaluate the fade detection accuracy, a statistical detection theory was used with Precision ($P$), Recall ($R$), and $F_1$ score [30]. $N_{TP}$ represents true positive, which is the ground truth (GT) and detected correctly. $N_{FP}$ is true negative, which is not GT but detected correctly. $N_{FN}$ is false negative, which is GT but not detected correctly. We calculated the $P$, $R$, and $F_1$ score as evaluation metrics. $P$ is defined as the ratio of correct experimental detections over the number of all true detections. $R$ is defined as the ratio of correct experimental detections over the number of all experimental detections. $F_1$ score is the harmonic mean of $P$ and $R$:

$$P = \frac{N_{TP}}{N_{TP} + N_{FP}},$$

$$R = \frac{N_{TP}}{N_{TP} + N_{FN}},$$

$$F_1 = 2 \cdot \frac{P \cdot R}{P + R}.$$  (15)

First, we compared the fade detection accuracy of the previous fade detection algorithms [26], [27], [29], [31]–[33] with the proposed fade detection algorithm using the TRECVID 2006 dataset (Table 1). Second, we evaluated the detection accuracy of the proposed fade detection algorithms using high-resolution video sequence which is composed of several movie trailer video listed in Table 4. Finally, we visually evaluated the $F_1$ score of the previous fade detection algorithms and the proposed fade detection algorithm.

A. EXPERIMENTAL ENVIRONMENT

Our certified database consists of the video sequences [24], which are listed in Table 1. A total of 476,593 frames were used for experiments. In addition we collected the datasets from eight movie trailer video sequences listed in Table 4. We also generated a ground truth for experiments, where the ground truth is that each frame is whether fade scene or not. The video sequences of dataset 1 were used to train and select the parameters because these sequences contained diverse types of fade scene, e.g., fade with white frame, containing letter box, roughly generated monochromatic frame, and light flipping. The rest of the sequences were used as test sequences. For the letterbox removal top and bottom pixels are excluded if all the pixels on same row has same pixel values. After the letterbox removal, resolution is downsampled into $320 \times 240$, determined experimentally. The three previous fade detection algorithms [26], [27], [29] are selected by the representative work that independently detect fade. In addition, to verify the performance of the proposed algorithm, we compared the performance of the deep learning-based fade detection algorithms [31]–[33]. In our experiment, various parameters used in the proposed method were also optimized and were set to values based on various experiments. We set threshold values of angle ($\theta_{th}$), curvature ($\kappa_{th}$), luminance transition tendency ($LTT_{th}$), and tendency similarity ($TS_{th}$) to 30°, 2.0, 0.5, and 0.9, respectively. Those threshold values are determined experimentally.

![FIGURE 7. Comparison of $F_1$ scores for different fade detection methods.](image)

![FIGURE 8. Precision/Recall measure of performance on the TRECVID 2006 dataset: (a) frame-based and (b) transition-based.](image)

B. PERFORMANCE EVALUATION OF THE PROPOSED METHOD

We evaluated the performance of the proposed fade detection algorithm. The test video sequences were classified by existing methods for comparison. The highest detection rates of the proposed method and those of the other methods are shown in boldface in Table 2 and Table 3. The proposed method achieved much higher accuracy than the other methods in terms of the $P$, $R$, and $F_1$ score.

We compared the proposed method and the existing methods on the TRECVID2006 dataset [24], which is a certified dataset. In Fig. 7, each existing method is represented by its corresponding reference number: [26], [27], [29], [31], [32], and [33]. In comparison, $R$ and $P$ indicate how accurately fade were detected at the frame level, as shown in Table 2.
Table 3 lists the detection accuracies at the transition level, not the frame level. In Tables 2 and 3, the best accuracy values are shown in boldface. The proposed method for frame-based and transition-based evaluations averaged 4.1% and 5.3% higher than the existing methods in terms of the $F_1$ score, respectively.

The accuracy shown in bold in Table 2 and Table 3 is the best accuracy. The proposed algorithm shows the best detection accuracy in most recalls and precision. However, for some video sequences, the proposed algorithm shows lower detection accuracy than existing algorithms. The causes of precision drop in some test sequences are as follows. In the scene change that occurs in low quality of video, it is observed that an afterimage of the previous frame remains despite the abrupt scene change. If it is recognized that the shot has changed over several frames, the precision of the proposed algorithm that defines the scene change based on the overall change pattern will be lowered. Conversely, the causes of the phenomenon of low recall are as follows. We reduce the number of falsely detected fades rather than find the correct fade start and end frames. For this reason, in the case of a video sequence with few fade scenes itself, the recall is lowered in the frame unit. Thus, in the detection of the scene change algorithm, we compare the $F_1$ score which is a harmonic mean of recall and precision, because recall and precision have a trade-off.

Fig. 7 shows a graph comparing the $F_1$ score calculated as the harmonic mean of $R$ and $P$. In Fig. 6, the proposed method...
shows an improved accuracy of at least 4.8% compared to the existing methods in terms of $F_1$ score. Specifically, the proposed method’s average $F_1$ scores are 0.195 (22.1%), 0.234 (26.5%), 0.309 (35.0%), 0.043 (4.8%), 0.134 (15.2%), and 0.206 (23.3%), which are higher than those of existing methods. Each improvement indicates the difference in value (ratio increase) between the existing and proposed method, respectively.

Fig. 8 shows the Precision/Recall measure of performance on the TRECVID 2006 dataset for each method based on the frame and transition, as shown in Tables 2 and 3. From Fig. 7, the proposed method achieved the best detection accuracy compared to all existing methods.

The $P$, $R$, and $F_1$ score of the proposed algorithm were measured for movie trailer videos that contain many fade scenes. The results are listed in Table 4. From the results in this table, the proposed method shows very high detection accuracy.

C. PERFORMANCE COMPARISON BY PARAMETER
The parameters used in the proposed algorithm are sensitive to slight changes. This is known as parameter sensitivity. Depending on the set values, which change the standard for the evaluation, the measured detection accuracy could be changed. The parameters to measure sensitivity are as follows: angle ($\theta_{th}$), curvature ($\kappa_{th}$), luminance transition tendency ($LTT_{th}$), and tendency similarity ($TS_{th}$). We conducted experiments using the test video sequences listed in Table 1, each modifying one parameter and leaving the others unchanged.

Fig. 9 shows the experimental results for the parameter sensitivity. In the case of $\theta_{th}$ and $LTT_{th}$, they show optimal performance at a specific value, while $\kappa_{th}$ and $TS_{th}$ show almost constant accuracy, and the performance decreases rapidly. In addition, because $P$ and $R$ show a trade-off as a whole, these parameters were set to maximize $F_1$ score. In the following experiments, the values that maximized $F_1$ score were 30°, 2.0, 0.5, and 0.9 for $\theta_{th}$, $\kappa_{th}$, $LTT_{th}$, and $TS_{th}$, respectively.

D. CONTRIBUTION DEMONSTRATION FOR EACH PROPOSED STEP
To verify the performance of each process in the proposed algorithm, each process was skipped separately, and the
accuracy was measured. The frame simplification is a process that prevents letterboxing and captions from interfering with the fade detection algorithm. Luminance flipping detection excludes the light source effect from fade detection. The measurements of the detection accuracy excluding those two processes are listed in Table 5. The transition boundary detection process cannot be skipped because the process searches the intervals of the candidates.

In Table 5, the detection accuracy for each of the proposed methods is shown. The proposed methods are divided into two categories: frame-based methods and luminance transition methods. Frame-based methods include frame simplification, luminance flipping detection, and dissolve detection. Luminance transition methods include luminance flipping detection and luminance transition detection.

VI. CONCLUSION

This paper proposed a new fade detection algorithm using the fluctuation of the luminance transition vector and luminance flipping detection. The existing fade detection algorithm has a limitation in that the precision is lowered by the dissolve scene and the flashing scene. Therefore, we consider dissolve and flashing scenes as the main challenge scenes and propose an algorithm to accurately distinguish fade scenes from them. In the proposed algorithm, the frame simplification process prevents misdetection caused by letterboxing or captions in the video, which are artificial edit effects that exist regardless of scene change. In addition, to prevent frame missing in frame-based fade detection algorithms, fade intervals are clarified with luminance transition vectors. Finally, luminance flipping detection eliminates the incorrect detection of fades due to flashing and light source movement in scenes, which are difficult to distinguish because they show luminance transition similar to fade scenes. The benefits of the proposed method were evaluated in terms of detection accuracy obtained from various video sequences. The experimental results demonstrated that the average $F_1$ of the proposed method was 0.187 (21.2% improvement) higher than those of existing methods.

SUPPLEMENTARY MATERIAL

Supplementary data associated with this article can be found online at http://csdl.postech.ac.kr/downloads/shyoon/Data.zip.

REFERENCES

[1] S. Abdulhussain, A. Ramli, M. Saripan, B. Mahmmmod, S. Al-Haddad, and W. Jassim, “Methods and challenges in shot boundary detection: A review,” Entropy, vol. 20, no. 4, p. 214, Mar. 2018.

[2] S.-J. Kang, “Adaptive luminance coding-based scene-change detection for frame rate up-conversion,” IEEE Trans. Consum. Electron., vol. 59, no. 2, pp. 370–375, May 2013.

[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, S. I. Cho, S. Yoo, and Y. H. Kim, “Multi-histogram based scene change detection for frame rate up-conversion,” in Proc. IEEE Int. Conf. Consum. Electron. (ICCE), Jan. 2013, pp. 332–333.

[5] B. Shahraray, “Scene change detection and content-based sampling of video sequences,” Proc. SPIE, vol. 2419, pp. 2–13, Apr. 1995.

[6] B.-L. Yeo and B. Liu, “Rapid scene analysis on compressed video,” IEEE Trans. Circuits Syst. Video Technol., vol. 5, no. 6, pp. 533–544, Dec. 1995.

[7] D. M. Thoumajoam, T. Khelchandra, K. M. Singh, and S. Roy, “A genetic algorithm and fuzzy logic approach for video shot boundary detection,” Comput. Intell. Neurosci., vol. 2016, pp. 1–11, Feb. 2016.

[8] J. Mas and G. Fernandez, “Video shot boundary detection based on color histograms,” in Proc. TRECVID, 2003, pp. 1–11.

[9] X. Qian, G. Liu, and R. Su, “Effective fades and flashlight detection based on accumulating histogram difference,” IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 10, pp. 1245–1258, Oct. 2006.

[10] O. Kucukten, U. Gu dukbay, and O. Ulusoy, “Fuzzy color histogram-based video segmentation,” Comput. Vis. Image Understand., vol. 114, no. 1, pp. 125–134, Jan. 2010.

[11] N. J. Janwe and K. K. Bhoyar, “Video shot boundary detection based on JND color histogram,” in Proc. IEEE 2nd Int. Conf. Image Inf. Process. (ICIIP), Dec. 2013, pp. 476–480.

[12] Z. Li, X. Liu, and S. Zhang, “Shot boundary detection based on multilevel difference of colour histograms,” in Proc. 1st Int. Conf. Multimedia Image Process. (ICMIP), Jun. 2016, pp. 15–22.

[13] R. Zubih, J. Miller, and K. Mai, “A feature-based algorithm for detecting and classifying scene breaks,” in Proc. 3rd ACM Int. Conf. Multimedia, 1995, pp. 189–200.

[14] J. Nam and A. H. Tewfik, “Combined audio and visual streams analysis for video sequence segmentation,” in Proc. IEEE Int. Conf. Aoust., Speech, Signal Process., vol. 4, Apr. 1997, pp. 2665–2668.

[15] R. W. Litz, “Comparison of automatic shot boundary detection algorithms,” Proc. SPIE, vol. 3656, pp. 290–301, Dec. 1998.
SANGHO YOON (Student Member, IEEE) received the B.S. degree in electrical engineering from the Pohang University of Science and Technology, Pohang, South Korea, in 2015, where he is currently pursuing the M.S. and Ph.D. degrees in computer vision and artificial intelligence.

YOUNG HWAN KIM (Life Senior Member, IEEE) received the B.S. degree in electrical engineering from the University of California at Berkeley, in 1985, and the M.S. and Ph.D. degrees in electrical engineering from the University of California at Berkeley, in 1988, respectively. From 1983 to 1988, he was a Postgraduate Researcher with the Electronic Research Laboratory, University of California at Berkeley, where he was involved in developing VLSI CAD programs. He is currently a Professor with the Division of Electronic and Computer Engineering, POSTECH, South Korea. His research interests include plasma display panel and liquid crystal display systems, multimedia circuit design, MPSoC and GPGPU system design for display and computer vision applications, and the development and commercialization of multimedia SoC. He was with the Department of Electrical Engineering, Ulsan National Institute of Science and Technology, Ulsan, South Korea, from 2001 to 2008, where he was involved in various military research projects, including the development of atherosclerosis detection and control systems. From 2008 to 2016, he served as the General Chair and a Committee Member for various Korean domestic and international technical conferences, including the International SoC Design Conference, the IEEE ISCAS 2012, and the IEEE APCCAS 2016. He has served as an Editor for the Journal of the Institute of Electronics Engineers of Korea.

SEOKHEYONG KANG (Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from the Pohang University of Science and Technology, Pohang, South Korea, in 1999 and 2001, respectively, and the Ph.D. degree in computer engineering from the University of California at San Diego (UCSD), La Jolla, in 2013. He was with the System-on-Chip (SoC) Development Team, Samsung Electronics, Suwon, South Korea, from 2001 to 2008, where he was involved in the development and commercialization of multimedia SoC. He was with the Department of Electrical Engineering, Ulsan National Institute of Science and Technology, Ulsan, South Korea, from 2014 to 2018. Since 2018, he has been with the Department of Electrical Engineering, Pohang University of Science and Technology. His current research interests include low power design optimization and cost-driven methodology for chip implementation.

CHANHEE LEE (Graduate Student Member, IEEE) received the B.S. degree in electrical and electronic engineering from Kyungpook National University, Republic of 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 at Daegu University, Republic of Korea. His current research interests include image analysis, computer vision, and circuit design for display and multimedia systems.

HO SUB LEE (Member, IEEE) received the B.S. degree in electrical and electronic engineering from Kyungpook National University, Republic of Korea, in 1977, and the M.S. and Ph.D. degrees in electrical engineering from the University of California at Berkeley, in 1985 and 1988, respectively. From 1977 to 1982, he was with the Agency for Defense Development, South Korea, where he was involved in various military research projects, including the development of atherosclerosis detection and control systems. From 1983 to 1988, he was a Postgraduate Researcher with the Electronic Research Laboratory, University of California at Berkeley, where he was involved in developing VLSI CAD programs. He is currently a Professor with the Division of Electronic and Computer Engineering, POSTECH, South Korea. His research interests include plasma display panel and liquid crystal display systems, multimedia circuit design, MPSoC and GPGPU system design for display and computer vision applications, statistical analysis and design technology for deep-submicron semiconductor devices, and power noise analysis. He has served as the General Chair and a Committee Member for various Korean domestic and international technical conferences, including the International SoC Design Conference, the IEEE ISCAS 2012, and the IEEE APCCAS 2016. He has served as an Editor for the Journal of the Institute of Electronics Engineers of Korea.

W. J. Heng and K. N. Ngan, “An object-based shot boundary detection using edge tracing and tracking,” in Proc. Int. Conf. Image Process., vol. 3, 1999, pp. 289–293.

S. Yoon and X. Liu, “A re-examination of text categorization methods,” in Proc. 8th ACM Int. Conf. Multimedia, 2000, pp. 217–219.

J. Zheng, F. Zou, and M. Shi, “An efficient algorithm for video shot boundary detection,” in Proc. Int. Symp. Intel. Multimedia, Video Speech Process., 2004, pp. 266–269.

M. Gygli, “Ridiculously fast shot boundary detection with fully convolutional neural networks,” in Proc. 22nd Annu. Int. Conf. Res. Develop. Inf. Retr. (SIGIR), vol. 12, no. 3, pp. 217–239, Sep. 2001.

S. Porter, M. Mirmehdi, and B. Thomas, “Temporal video segmentation and classification of edit effects,” Image Vis. Comput., vol. 21, nos. 13–14, pp. 1097–1106, Dec. 2003.

B. T. Truong, C. Dorai, and S. Venkatesh, “New enhancements to cut, fade, and dissolve detection processes in video segmentation,” in Proc. 8th ACM Int. Conf. Multimedia, 2000, pp. 217–219.

J. Z. Miadowsicz, “Story tracking in video news broadcasts,” Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Univ. Kansas, Lawrence, KS, USA, 2004.

P. Browne, A. F. Smeaton, N. Murphy, N. E. O’Connor, S. Marlow, and C. Berrut, “Evaluating and combining digital video shot boundary detection algorithms,” Tech. Rep., 2000.

G. C. Chávez, F. Precioso, M. Cord, S. Philipp-Foliguet, and A. de Albuquerque Araújo, “Shot boundary detection at TRECVID 2006,” in Proc. TRECVID, 2006, pp. 1–11.

M. C. Lupo, A. F. Smeaton, and J. McCallum, “A re-examination of video shot boundary detection through spatio-temporal convolutional neural networks,” in Proc. Int. Conf. Multimedia Indexing (CBMI), Sep. 2018, pp. 1–4.

M. H. Lee and S. M. Park, “Fast video shot transition localization with deep structured models,” in Proc. Asian Conf. Comput. Vis. (ACCV), 2018, pp. 577–592.