IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received December 2, 2019, accepted December 15, 2019, date of publication December 25, 2019,
date of current version January 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2962328

A Novel Shot Detection Approach Based on ORB Fused With Structural Similarity

HUIBIN LIU, TAN-HSU TAN, (Senior Member, IEEE), AND TIEN-YING KUO, (Member, IEEE)
Department of Electrical Engineering, National Taipei University of Technology, Taipei 10608, Taiwan

Corresponding author: Huibin Liu (t106319405@ntut.edu.tw)

This work was supported by the Ministry of Science and Technology of Taiwan under Grant MOST 107-2221-E-027-106.

ABSTRACT Shots are the basic units for analyzing and retrieving video, and also the essential elements in creating video datasets. The traditional methods of shot detection exhibit unsatisfactory performance for being too sensitive to motion or too much time-consuming. This paper proposes an automatic shot detection method, by employing the fast feature descriptor of Oriented FAST and Rotated BRIEF (ORB) fused with Structural Similarity (SSIM). Firstly, ORB descriptor is used to preselect candidate segments with a high tolerance for rapidly extracting the features of twenty-frame intervals in video sequences. Then, the cut transition is detected by comparing ORB features, fused with SSIM, of consecutive frames in the candidate segment. Finally, the gradual transition is detected by determining the maximum amount of the continuous increasing/decreasing interframe differences in the candidate segment without cut transition. Experimental result indicates that the proposed method can achieve an F1-Score of 92.5% and five times of real-time speed with one CPU on 106049 test frames from the Open-video project, YouTube, and YOUKU. In addition, the proposed method can outperform the existing shot detection methods, including the rule-based and learning-based methods, by testing on the video sequences from the Open-video project and RAI dataset.

INDEX TERMS Shot detection, SSIM, ORB descriptor, cut transition, gradual transition.

I. INTRODUCTION

Techniques for intelligent service based on big data have been rapidly developed, and the video information plays an important role in those systems, such as autopilot system [1], [2] and smart city [3]–[5]. This leads to video information overload. Therefore, effectively organizing, extracting and querying visual information need to be solved urgently [6]–[9]. Content-based video retrieval has attracted a lot of attention recently [10], [11], especially moving from theory to practice due to the advances of large structured multimedia datasets. Shots are the basic units of video, which can serve as the basis for the succeeding video content analysis and retrieval. Before being interpreted, stored and retrieved based on content, video data should be segmented effectively. Therefore, shot detection is an important part for video information analysis [12]. There are two types of transitions between shots: CT (Cut Transition) and GT (Gradual Transition). CT is an abrupt transition from one shot to another. GT may have many forms, such as dissolve, fade-in/out, and wipe. During the past decades, many shot detection algorithms have been presented [13]–[15] to detect CT as well as GT. The traditional rule-based shot detection methods are divided into two typical categories: shot detection methods based on color and shot detection methods based on texture.

Color is the first visual character being used in shot detection [16], [17]. Many color-based shot detection methods have been developed. Generally, the simplest way is to use the absolute value of the histogram to measure the distance between two consecutive frames to realize shot detection. Some other color-based methods which calculate the distance, such as accumulated histogram, center distance in color space HSI or MTM, or the weighted distance of histogram, intersecting method etc., have also been proposed. In [17], the normalized HSV color histogram is utilized to detect the differences between frames and the SVD is performed to speed up shot detection. Nevertheless, the color-based shot detection methods can easily result in error shot detection: entirely different frames with similar histograms are mistaken for belonging to the same shot, because of losing the local information.

As for the texture-based approach, the texture features of frames are extracted by a co-occurrence matrix based on space information, and obtained by computing a joint

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

frequency a distribution of two gray-scale pixels with a distance ($\triangle x$, $\triangle y$). A variety of gray co-occurrence matrix statistics can be used to measure the texture features, including angular second moment, entropy, correlation, etc. Those features will be normalized to calculate the distance between two frames [18]. A hybrid shot detection method is presented by using texture, color, edge, and motion as feature vectors to achieve excellent accuracy [19]. However, the computational burden is very high. Other traditional shot detection methods include methods based on shape [20] and spatial relations. Actually, several different features are usually fused to calculate the difference between frames. Nevertheless, each of these methods demonstrates unsatisfactory performance for being sensitive to motion, having high computational cost or lacking local information.

In recent years, some learning-based methods are proposed to detect transitions via convolutional neural network (CNN). [21] presents a shot detection technique based on spatio-temporal CNN architecture, which analyzes both spatial and temporal information of video sequences through CNN, with a large training dataset containing more than 3.5 million frames. Similarly, [22] generates a transition dataset with 1 million frames and proposes a shot detection CNN model, which runs the input frames through 3D convolutions and is fully convolutional in time. However, there are two significant limitations in learning-based methods: running on expensive GPU and constructing the large-scale training dataset. In other words, the efficiency and accuracy of learning-based shot detection methods are largely determined by the performance of graphics hardware and the size of the training dataset.

To reduce computational burden and overcome the hardware constraints as mentioned above, this study presents a new shot detection method by integrating Oriented FAST and Rotated BRIEF (ORB) [23] descriptor and Structural Similarity (SSIM) [24]. ORB descriptor is used to extract the candidate segments rapidly and coarsely with a much less computational burden. Cut transition and gradual transition are successively detected in each candidate segment by matching ORB and SSIM features of corresponding consecutive frames. Experiments on video sequences from Open-video project [15], [25], RAI dataset [26], YouTube and YOUKU show that the proposed method outperforms the existing shot detection methods. The main contribution of this study can be summarized as follows:

- *Achieving high accuracy as well as efficiency by matching ORB features fused with SSIM in twenty-frame intervals.*
- *Designing a GT model based on the regularity of gradual transition between two shots.*
- *Presenting the look-back mechanism to improve the precision rate for both CT and GT detection.*

II. FUSED SHOT DETECTION METHOD

The proposed shot detection method consists of three modules: the candidate segment preselection module, the cut transition detection module, and the gradual transition detection module. The candidate segment preselection module is responsible for quickly selecting candidate segments that may transition, the latter two modules are responsible for detecting cut transition and gradual transition, respectively. If the cut transition detection module fails to obtain cut transition, the gradual transition detection is involved to detect gradual transition. Otherwise, the program goes back to the candidate segment preselection module directly. As depicted in the upper part of Fig. 1, ORB descriptor is utilized to preselect the candidate segments in video sequences by employing the candidate segment preselection module, which can extract the features from twenty-frame intervals very fast. Then, in the following cut transition detection module, CT is detected by comparing ORB features fused with SSIM of consecutive frames in the candidate segment. If there is no CT, the last module in Fig. 1 will be employed to detect GT in the candidate segment.

A. CANDIDATE SEGMENT PRESELECTION MODULE

It is reasonable to suppose that there is at most one transition in twenty-frame intervals of the video sequence. Accordingly, the proposed method takes 20 frames as a segment to preselect candidate segments, which may include one CT or one GT. In the candidate segment preselection module, ORB descriptor is used to extract the feature of the last frame of 20 frames in each segment. ORB [23] is a fast and robust visual feature detector, which is based on FAST keypoint detector [27] and BRIEF feature descriptor [28]. ORB descriptor is widely used in image matching [29] and SLAM systems [30], [31].

The proposed candidate segment preselection module is performed with the following four steps: (1) extracting feature, (2) finding two nearest match points for each keypoint, (3) counting good match points, and (4) selecting candidate segments.

(1) Extracting feature: the feature of the last frame of each segment is extracted via ORB descriptor. A large number of key points and their feature descriptions are obtained by the fast detector.

(2) Finding two nearest match points: k-nearest neighbors algorithm is used to find two nearest match points ($kp'_1$and kp'_2) from the last frame of the previous segment for each keypoint (kp) in the current frame.

(3) Counting good match points: the number of good match points is counted by calculating the ratio of the two nearest match points. The number of good match points is counted based on each keypoint. For each keypoint, it is counted as follows:

$$N_{FF'} = \begin{cases} N_{FF'} + 1 & \dfrac{dis1}{dis2} < 0.75 \\ N_{FF'} & otherwise \end{cases} \tag{1}$$

where F and F' are respectively the last frames of the current segment and the previous segment. $N_{FF'}$ is the number of

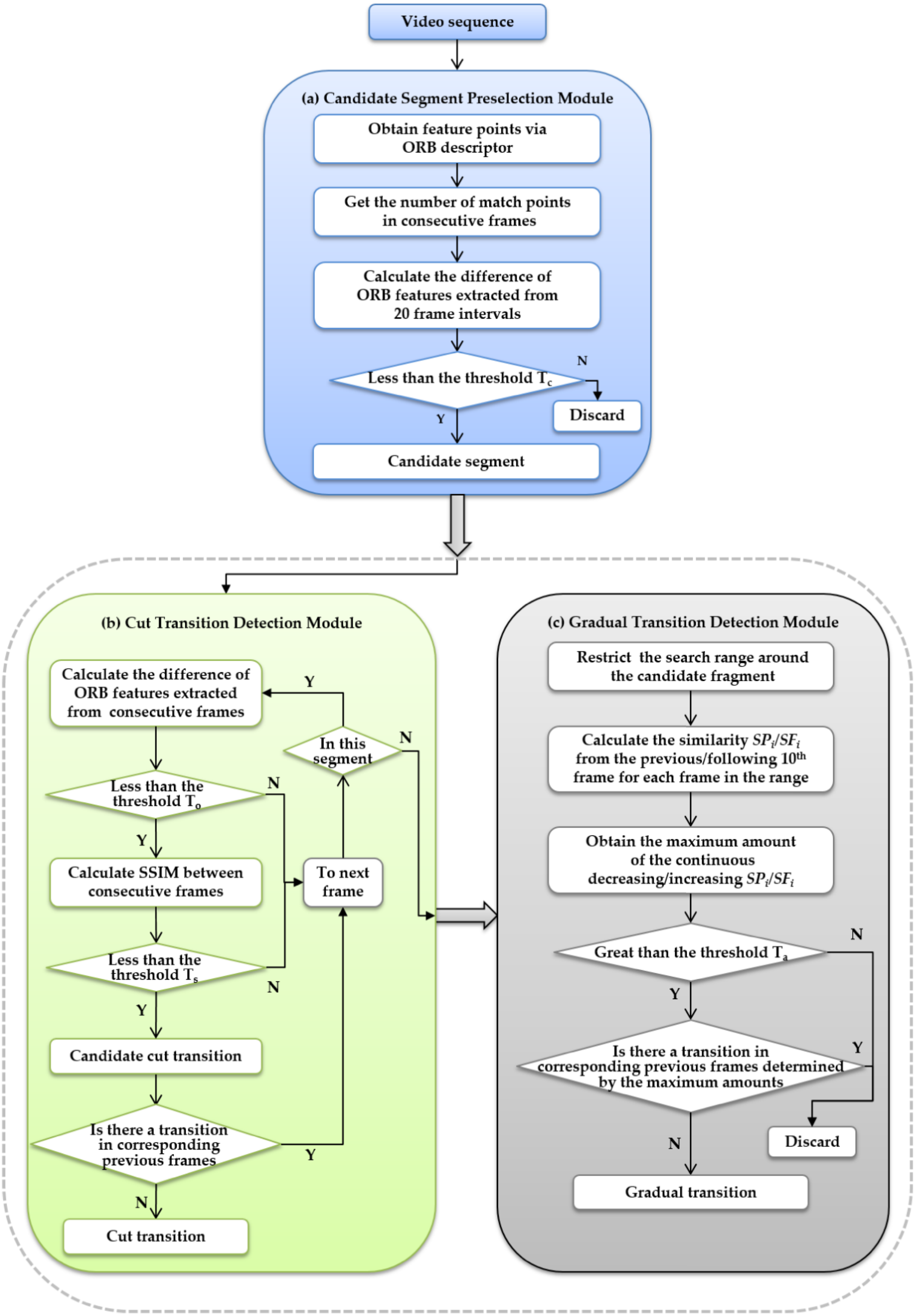
Video sequence
(a) Candidate Segment Preselection Module
Obtain feature points via ORB descriptor
Get the number of match points in consecutive frames
Calculate the difference of ORB features extracted from 20 frame intervals
Less than the threshold T_c
N
Discard
Y
Candidate segment
(b) Cut Transition Detection Module
Calculate the difference of ORB features extracted from consecutive frames
Y
In this segment
N
Less than the threshold T_o
N
Y
Calculate SSIM between consecutive frames
To next frame
Less than the threshold T_s
N
Y
Candidate cut transition
Is there a transition in corresponding previous frames
Y
N
Cut transition
(c) Gradual Transition Detection Module
Restrict the search range around the candidate fragment
Calculate the similarity SP_i/SF_i from the previous/following 10th frame for each frame in the range
Obtain the maximum amount of the continuous decreasing/increasing SP_i/SF_i
Great than the threshold T_a
N
Y
Is there a transition in corresponding previous frames determined by the maximum amounts
Y
Discard
N
Gradual transition

FIGURE 1. Execution flowchart of the fused shot detection method. (a) Candidate segment preselection module, (b) Cut transition detection module, and (c) Gradual transition detection module.

good match points from two frames in 20-frame intervals, $dis1$ is the distance between kp and the best nearest match point kp'_1, and $dis2$ is the distance between kp and the second-best nearest match point kp'_2.

(4) Selecting candidate segments: if $N_{FF'}$ is less than the threshold T_C, the current segment is selected as one of the candidate segments. A large number of segments without transition might be discarded in this step.

The candidate segment preselection threshold T_C is relative to the number of keypoints extracted via ORB and the resolution of video sequence. When the number of keypoints and the resolution of video sequences are equal to 300 and 320 × 240, respectively, the value of T_C should be small enough to consider the frames with few corners and is set to 10 empirically. After preselection, the candidate segments with a potential cut transition or gradual transition are selected from the video sequence. Meanwhile, ORB descriptor will produce a certain amount of wrong detected segments with fast motion, which is mistaken as a transition. In the following modules, cut transition and gradual transition are successively detected in each candidate segment. The following pseudo code (Algorithm 1) gives the brief implementation of the candidate segment preselection module.

Algorithm 1 Candidate Segment Preselection

```
$frame0 = Startframe$;
$ORB\_descriptor(frame0)$;
for $frame = Startframe + 20$; $frame < Endframe$; $frame+ = 20$ do
    $ORB\_descriptor(frame)$;
    $N_{FF'} = Match(frame0, frame)$;
    if $N_{FF'} < T_C$ then
        $CT\_detection(frame)$;
    end if
    $frame0 = frame$;
end for
```

B. CUT TRANSITION DETECTION MODULE

By comparing the eigenvalues of two consecutive frames in each candidate segment extracted by ORB descriptor, the first frame of each potential cut transition which is considered to be the beginning of a shot is obtained, while the quantity of good match points in its previous frame is less than the threshold T_O. Similar to T_C, T_O is relative to the number of keypoints extracted via ORB and the resolution of the video sequence, which is set to 45 when the number of keypoints and the resolution are equal to 300 and 320×240. However, a certain amount of wrong detected cut transitions may appear simultaneously. Because the visual feature detected by ORB is based on keypoints, the continuous dark frames are regarded as new cut transitions as shown in Fig. 2*a*. Another case is that there are big foreground objects in the video sequence. When the objects are moving in consecutive frames as shown in Fig. 2*b*, the proposed cut transition detection algorithm has failed because the pair of keypoints could not match well. Therefore, the detected cut transitions are refined by applying SSIM to remove the wrong detected cut transitions.

To overcome the difficulty mentioned above, SSIM is introduced to address the first frames of the potential cut transitions detected by calculating the difference of ORB features between the consecutive frames. SSIM [24] is a novel image quality assessment approach for evaluating the difference between the reference and distorted images. In this study, SSIM is employed to measure the structural similarity between the first frame of each detected CT and its previous frame.

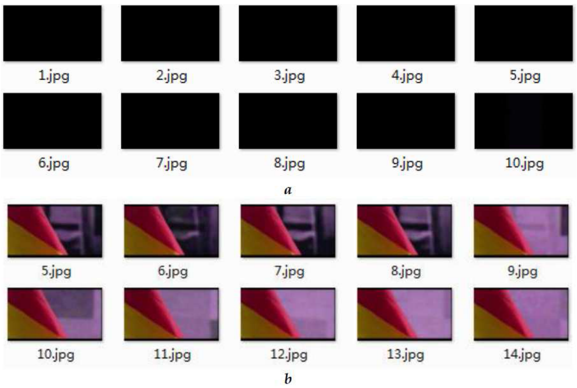

FIGURE 2. Wrong detected cut transitions by ORB descriptor.(a)Cut transitions detected in the continuous dark frames,(b)Cut transitions detected in the consecutive frames with big foreground object.

SSIM consists of three items: luminance comparison, contrast comparison, and structure comparison. The luminance comparison between the two consecutive frames is defined by:

$$l\left(f, f''\right) = \frac{2\mu_f \mu_{f''} + C_1}{\mu_f^2 + \mu_{f''}^2 + C_1} \tag{2}$$

where f is one of the local windows (blocks) of the first frame in the potential CT detected by ORB and f'' is the corresponding window in the previous frame of the first frame. The constant C_1 is used to avoid zero denominators. The same rule holds in contrast comparison and structure comparison [23]. μ_f and $\mu_{f''}$ are the mean intensities of the first frame and the previous frame, respectively. μ_f is defined as below:

$$\mu_f = \frac{1}{N}\sum_{i-1}^{N} f_i \tag{3}$$

f_i is the i^{th} pixel of the local window. N is the number of pixels in the local window. In [23], it is verified that the local SSIM index is superior to global SSIM index.

Next, the contrast comparison between the two corresponding windows in the consecutive frames is defined by:

$$c\left(f, f''\right) = \frac{2\sigma_f \sigma_{f''} + C_2}{\sigma_f^2 + \sigma_{f''}^2 + C_2} \tag{4}$$

where σ_f is the standard deviation and employed as an estimate of the contrast:

$$\sigma_f = \left(\frac{1}{N-1}\sum_{i=1}^{N}\left(f_i - \mu_f\right)^2\right)^{\frac{1}{2}} \tag{5}$$

Further, the structure comparison between the two corresponding windows in the consecutive frames is defined as:

$$s\left(f, f''\right) = \frac{\sigma_{ff''} + C_3}{\sigma_f \sigma_{f''} + C_3} \tag{6}$$

where $\sigma_{ff''}$ is the covariance of f' and f'', which is utilized to measure the structure similarity, and defined as below:

$$\sigma_{ff''} = \frac{1}{N-1}\sum_{i=1}^{N}\left(f_i - \mu_f\right)\left(f_i'' - \mu_{f''}\right) \tag{7}$$

To integrate the three items, eqs. (2), (4) and (6) are combined, with $C_3 = C_2/2$. Further, the structure comparison between the two corresponding windows in the consecutive frames is defined as:

$$SSIM\left(f, f''\right) = l\left(f, f''\right) \cdot c\left(f, f''\right) \cdot s\left(f, f''\right) \tag{8}$$

$$SSIM\left(f, f''\right) = \frac{\left(2\mu_f \mu_{f''} + C_1\right)\left(2\sigma_{ff''} + C_2\right)}{\left(\mu_f^2 + \mu_{f''}^2 + C_1\right)\left(\sigma_f^2 + \sigma_{f''}^2 + C_2\right)} \tag{9}$$

Finally, the mean of SSIM is used to evaluate the overall frame similarity as below.

$$MSSIM_{FF''} = \frac{1}{M}\sum_{j=1}^{M} SSIM\left(f_j, f_j''\right) \tag{10}$$

where F and F'' are respectively the first frame in the potential cut transition and its previous frame, and M is the number of local windows in the current frame.

As mentioned above, CT is firstly detected by comparing ORB features between two consecutive frames in the candidate segments. If the number of good match keypoints in the consecutive frames is less than the threshold T_O, structure similarity between the consecutive frames will be calculated to refine the result of CT detection. Here, the threshold of structure similarity, T_S, is set to 0.7 to achieve high recall rate with a higher tolerance. In order to improve precision rate, the confidence coefficient of the current frame F (CC_F) is developed by calculating the differences between the actual values and the thresholds, and being normalized to range from 0 to 1. The confidence coefficient is defined by:

$$CC_F = \frac{T_O - N_{FF''}}{T_O} + \frac{T_S - MSSIM_{FF''}}{T_S} \tag{11}$$

where $N_{FF''}$ is the number of good match keypoints between the current frame and its previous frame. CC_F ranges from 0 to 2, which represents the confidence to be the first frame of CT for the current frame. Looking back in corresponding previous frames is considered based on CC_F. The purpose here is to avoid wrong detection when the difference between the current frame and its consecutive frame is caused by the previous transition. The look-back mechanism is defined as:

$$N_{lb} = \begin{cases} NULL & CC_F \leq 1 \\ 2 \times FR & 1 < CC_F \leq 1.5 \\ 1 \times FR & 1.5 < CC_F \leq 1.7 \\ 0 & otherwise \end{cases} \tag{12}$$

N_{lb} is the number of look-back frames for finding the existence of the previous transition. FR is the frame rate of the video sequence. A detailed description of the look-back mechanism is given below:

(1) If CC_F of the current frame is less than 1, it is determined that the current frame is not the first frame of CT, and N_{lb} need not be assigned.

(2) Otherwise, if CC_F is greater than 1.7, the current frame is identified as the first frame of a new shot directly. The rest of the fames in the segment are no longer considered.

(3) In other cases, the current frame is identified as the first frame of a new shot only when there is no transition in the previous N_{lb} frames.

As indicated in Fig. 1, the current frame is not identified as the first frame of CT in the following three cases: (1) $N_{FF''}$ is greater than T_O, (2) SSIM is greater than T_S, and (3) there is a transition in corresponding previous frames. After traversing all frames in the segment, if there is no CT, GT will be detected from the candidate segment in the following gradual transition detection odule. The following Algorithm 2 shows a pseudo code of the CT detection algorithm.

Algorithm 2 CT Detection

```
for $i = 1; i < 20; i++$ do
    $frame1 = frame + i$;
    $ORB\_descriptor(frame1)$;
    $N_{FF''} = Match(frame1 - 1, frame1)$;
    if $N_{FF''} < T_O$ then
        $M_{FF''} = MSSIM(frame1 - 1, frame1)$;
        if $M_{FF''} < T_S$ then
            if $Look\_back = False$ then
                $CT\ Exist$;
                $Break$;
            end if
        end if
    end if
end for
if $i = 20$ then
    $Gt\_detection(frame)$;
end if
```

C. GRADUAL TRANSITION DETECTION MODULE

Each candidate segment without CT may have potential GT. When the gradual transition occurs, the previous shot gradually changes to the next shot. The frames in GT have the features of both previous and next shots. In theory, the similarity with the previous shot is decremented frame by frame. At the same time, the similarity with the next shot is incremented frame by frame. In the proposed gradual transition detection module, the previous and following 10^{th} frames are employed to replace the previous and next shots.

Fig. 3 shows the gradual transition model, which illustrates the change rule of the structural similarities with the previous and following 10^{th} frames for each frame in GT. When GT occurs, structural similarities between the frames in the candidate segment and their previous 10^{th} frames have a continuous decreasing tendency. Meanwhile, structural similarities between the frames in the candidate segment and their following 10^{th} frames have a continuous increasing tendency

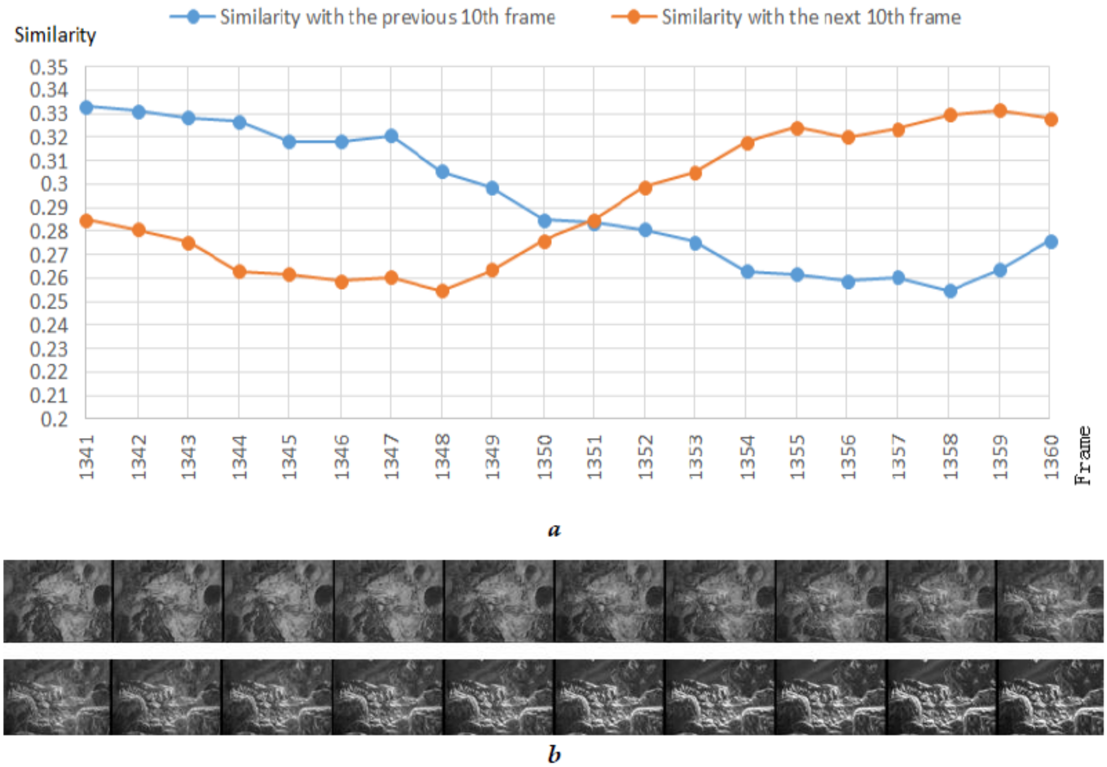

FIGURE 3. Gradual transition model.(a)Changing tendency of SSIM,(b)Frames in candidate segment.

as demonstrated in Fig. 3*a*. The first row in Fig. 3*b* exhibits Frames 1341 to 1350, and the second one shows Frames 1351 to 1360, which belong to the candidate segment with a gradual transition.

The proposed gradual transition detection module is performed with the following four steps: (1) expanding search range, (2) calculating SSIM, (3) obtaining maximum amount, and (4) looking back.

(1) Expanding search range: A gradual transition may happen across multiple segments. Therefore, before detecting a gradual transition, the search range is expanded by adding previous and following 10 frames of current candidate segment.

(2) Calculating SSIM: The similarity between each frame in the range and its previous/following 10^{th} frame, SP_i/SF_i, is calculated.

(3) Obtaining maximum amount: The maximum amounts of frames with the continuously decreasing SP_i and increasing SF_i are obtained simultaneously, after traversing all frames in the range.

(4) Looking back: Look-back mechanism is also considered based on the maximum amounts. If the maximum amounts are large enough, the existence of GT is determined directly. Otherwise, looking back to the corresponding previous frames is performed to ascertain a previous GT when both of the maximum amounts are greater than the amount threshold T_a. The GT detection algorithm (Algorithm 3) is simply described by the pseudo code as shown below.

Algorithm 3 GT Detection

```
for $i = 1; i < 20; i++$ do
    $frame2 = frame + i$;
    $M\_Previous\_Shot = MSSIM(frame2 - 10, frame2)$;
    $M\_Next\_Shot = MSSIM(frame2, frame2 + 10)$;
    $C\_d = Count(continuously\ decreasing\ M\_Previous\_Shot)$;
    $C\_i = Count(continuously\ increasing\ M\_Next\_Shot)$;
end for
if $Max(C\_d) > T_a$ and $Max(C\_i) > T_a$ then
    if $Look\_back = False$ then
        $GT\ Exist$;
    end if
end if
```

In order to improve the recall rate, ORB features of ten-frame intervals are extracted to detect GT in the search range when no GT is identified by comparing SSIM of ten-frame intervals. GT detection by use of ORB follows the same steps mentioned above.

III. RESULTS AND DISCUSSION

A. IMPLEMENTATION DETAILS

The performance of our method is compared with two rule-based shot detection algorithms [17], [19] and two learning-based shot detection algorithms [21], [22], in terms of Precision (P), Recall (R) and F-Measure ($F1$) [32]. In the

TABLE 1. Characteristics of nine video sequences used in the experiment.

Video	Frames	Transitions			Resolution	Sources
		Total	Cut	Gradual		
D1	11356	65	38	27	320×240	NASA 25th Anniversary, Airline Safety and Economy, Perseus Global Watcher
D2	16586	73	42	31	320×240	
D3	12304	103	39	64	320×240	
D4	31389	153	98	55	320×240	
D5	12508	71	45	26	352×240	
D6	13648	85	40	45	352×240	
Show	2494	38	38	0	426×240	YouTube
Cartoon	2256	23	21	2	720×406	YOUKU
News	3508	46	41	5	384×288	YOUKU

experiments, our method is evaluated on the video sequences from the Open-video project [15], [25], RAI dataset [26], YouTube and YOUKU, implementing in Python 3.7 with a 3.0 GHz CPU and 8 GB RAM.

The Precision, Recall, and F-Measure of each approach are calculated, respectively, by employing eq. (13).

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

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

$$F1 = \frac{2 \times P \times R}{P + R} \tag{13}$$

In this study, TP (True Positive) is the number of correctly detected shots. FP (False Positive) is the number of shots being mistaken for an independent shot which actually belongs to the previous shot. FN (False Negative) represents the number of shots that are missed detected. $F1$ is a combination of precision and recall.

B. COMPARISON WITH RULE-BASED METHODS

The test video sequences consist of six videos D1-D6 from the Open-video project [15], [25], show [33], cartoon [34], and news video (the beginning of [35]). Table 1 illustrates characteristics of the nine video sequences. The videos from the Open-video project have been used to validate the performance of methods in [17] and [19]. The other three videos are introduced to measure the accuracy of the proposed method on the videos with different resolution and frame rate.

Table 2 shows the comparative result of the proposed method with other two rule-based shot detection methods [17], [19] by testing on four video sequences, which have been reported in these two studies. As shown in Table 2, the proposed method outperforms the other two methods in four of six items, including the average values of P, R, $F1$ for cut transition and gradual transition. Especially for CT detection, the proposed method achieves an average F1-score of 95.0 percent. However, the precision values of both CT and GT detected by the proposed method on D2 video sequence are lower than those using other sequences. Due to the high sensitivity of SSIM to local change, the proposed method yields a certain number of wrong detections. Fig. 4 shows a wrong detected GT in consecutive frames which have an obvious local change in luminance, contrast, and structure caused by the flame flickering in the twenty-frame intervals. Furthermore, there are only a few corner features in these frames, leading to the failure of identification by ORB descriptor.

The work of [19] also gives the detection results on other two video sequences provided by the Open-video project. Table 3 illustrates the comparative result of detection results on overall transitions employing six test sequences between the proposed method and the method proposed in [19]. Compared with the method in [19], the average of all precision values for the proposed method is less than 0.9 percent. However, the recall average of the proposed method is greater than that of the method in [19] by 4.5 percent, and its F1 average is 1.6 percent greater. Although the performance of the proposed method is not significantly better than that of the method in [19], the computational cost of the proposed method is much lower, which will be discussed later. Table 4 presents processing time, P, R, $F1$ of the fused shot detection method testing on all video sequences listed in Table 1. Seven of nine precise values are greater than 90 percent, and the average is 92.2 percent. Eight of nine recall values are greater than 91 percent, and the average is 93.2 percent. The average value of F1 is 92.5 percent. Although the method in [17] can achieve an obvious high processing speed, approximately fifty times of real-time, its accuracy of shot detection is much lower than the proposed method. The method in [19] claimed that the processing speed of shot detection is nearly two times of real-time with 3.6GHz CPU. The third column in Table 4 shows faster processing speed of the proposed method, which is higher than five times of real-time. In summary, the proposed method outperforms the shot detection method of [19] in terms of speed and accuracy.

C. COMPARISON WITH LEARNING-BASED METHODS

In this section, the proposed method is compared with the learning-based shot detection methods in [21] and [22], which detect the transitions via CNN. Table 5 shows the comparative result of overall transition between the proposed method and the method in [21], by testing on the video sequences also from the Open-video project, which has been employed to evaluate the accuracy of shot detection in [21]. The average of all F1 values in the proposed method is 5 percent greater than [21]. Remarkably, the F1 score of video 6011 in [21] is only 30.5 percent, which may be caused by the absence of similar videos in the training dataset. In contrast, the F1 scores obtained by the proposed method are approximately equal to or greater than 85 percent except Video 50009, which has violent tremors frequently.

Table 6 indicates the comparative result of overall transition detection between the proposed method and the method in [22] on RAI dataset, which has been reported in [22]. We adopt the same expression as in [22]. The values of

TABLE 2. Comparative result of CT and GT detections with different methods using four test sequences.

Video	The method in [17]						The method in [19]						Proposed Method					
	CT			GT			CT			GT			CT			GT		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
D2	90.5	90.5	90.5	72.5	93.5	81.7	85.4	97.6	91.1	90.0	87.1	88.5	89.4	100.0	94.4	77.5	100	87.3
D3	86.7	66.7	75.4	94.0	73.4	82.4	86.5	82.1	84.2	88.7	85.9	87.3	97.4	94.9	96.1	91.1	79.7	85.0
D4	89.7	88.8	89.2	74.1	72.7	73.4	90.6	88.8	89.7	84.6	80.0	82.2	97.9	96.9	97.4	81.4	87.3	84.2
D6	97.4	95.0	96.2	92.7	84.4	88.4	97.4	95.0	96.2	88.5	88.5	88.5	97.2	87.5	92.1	76.8	95.6	85.1
Average	91.1	85.3	87.8	83.3	81.0	81.5	89.0	91.0	89.9	87.9	85.4	86.6	95.5	94.8	95.0	81.7	90.7	85.4

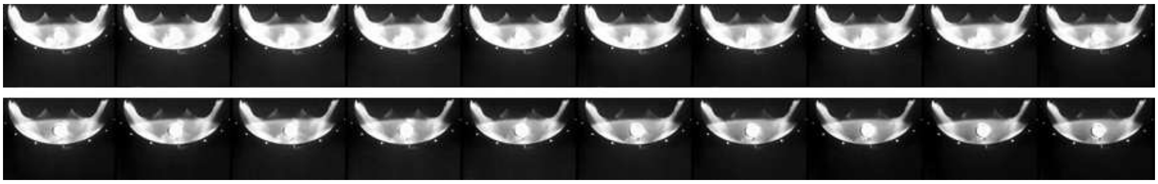

FIGURE 4. Wrong detected gradual transition with obvious local change.

TABLE 3. Comparative result of overall transition detection using six test sequences.

Video	The method in [19]			Proposed Method		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
D1	95.2	92.3	93.8	92.8	98.5	95.5
D2	87.2	93.2	90.1	83.9	100.0	91.3
D3	87.9	84.5	86.1	93.6	85.4	89.3
D4	88.5	85.6	87.0	91.8	93.5	92.6
D5	91.4	90.1	90.8	90.5	94.4	92.4
D6	92.8	90.6	91.7	84.8	91.8	88.1
Average	90.5	89.4	89.9	89.6	93.9	91.5

TABLE 4. Performance of the fused shot detection method testing on nine test sequences.

Video	Duration (m:s)	Time (s)	P(%)	R(%)	F1(%)
D1	06:19	66.7	92.8	98.5	95.5
D2	09:13	108.5	83.9	100.0	91.3
D3	06:50	81.2	93.6	85.4	89.3
D4	17:27	180.0	91.8	93.5	92.6
D5	07:06	71.9	90.5	94.4	92.4
D6	07:42	89.9	84.8	91.8	88.1
Show	01:23	16.4	94.6	92.1	93.3
Cartoon	01:30	24.2	100.0	91.3	95.5
News	03:53	39.6	97.7	91.3	94.4
Average	06:49	75.4	92.2	93.2	92.5

P, R, and F1 have no decimals. All of the P, R and F1 scores in our method are greater than 80 percent. However, there are some extremely low values below 70 percent in the results of [22]. It provides further evidence that the learning-based shot detection methods are not robust to unseen videos. Although the shot detection method in [22] runs at 121 × real-time, a high performance GPU is required.

TABLE 5. Comparative result of overall transition detection between the method in [21] and the proposed method.

Video	The method in [21]			Proposed Method		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
01811a	89.6	93.8	91.6	98.4	93.8	96.0
6011	28.9	32.2	30.5	88.1	91.7	89.9
8024	76.8	90.6	83.1	92.2	78.3	84.7
8386	95.8	96.6	96.2	83.5	89.8	86.5
8401	78.9	96.8	87.0	90.3	90.3	90.3
10558a	99.2	96.2	97.7	90.0	90.0	90.0
23585a	95.2	96.4	95.8	94.6	95.2	94.9
23585b	96.3	99.0	97.6	94.5	99.0	96.7
34921a	92.2	94.7	93.4	95.5	85.3	90.1
34921b	89.2	91.9	90.5	92.3	84.8	88.4
36553	88.6	94.4	91.4	93.7	96.7	95.2
50009	64.6	91.4	75.7	72.1	84.5	77.8
50028	83.2	95.7	89.0	94.7	96.8	95.7
UGS01	93.4	97.2	95.3	93.9	96.0	94.9
UGS04	93.7	99.6	96.5	93.2	97.8	95.4
UGS05	55.3	86.7	67.5	93.8	100.0	96.8
UGS09	91.2	91.2	91.2	91.8	92.7	92.3
Average	83.1	90.8	86.5	91.3	91.9	91.5

TABLE 6. Comparative result of overall transition detection between the method in [22] and the proposed method.

Video	The method in [22]			Proposed Method		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
21829	81	64	72	90	92	91
21867	89	84	86	84	84	84
23553	95	99	97	83	85	84
23557	91	97	94	94	100	97
23558	92	99	95	88	87	87
25008	94	94	94	86	85	85
25009	97	96	96	89	81	85
25010	93	94	93	99	91	95
25011	62	90	73	96	95	95
25012	66	89	76	98	96	97
average	86.0	90.6	87.7	90.7	89.6	90.1

D. DISCUSSION

In order to compare the efficiency and practicality of the proposed method with other existing shot detection methods, Table 7 describes the pros and cons of the compared methods. The proposed method has the advantages of other shot detection methods while avoiding their disadvantages. Although it is not the fastest detection method, it still achieves 5 × real-time running on CPU, and has the highest accuracy among the compared methods without the limitations of hardware. As shown in Table 7, a few false positives

TABLE 7. Comparative Analysis among the existing shot detection methods.

Method	Pros	Cons
SVD and Pattern Matching [17]	High detection speed	Lacks in detection accurancy, especially for gradual transition
Feature Vector [19]	High accuracy	Heavy optical flow computation
Spatio-temporal Convolutional Neural Networks [21]	Competitive performance and high detection speed	Requires GPU and large-scale training dataset
Fully Convolutional Neural Networks [22]	121 × real-time	Requires GPU and fails to unseen videos in the training dataset
The proposed method	High accuracy, device-independent and low computational burden	A few false positives caused by blur and flicker

are caused by blur and flicker that reduce the precision of the proposed method. As shown in Fig.4 above, a wrong detection may be caused by an obvious local change in luminance. This can be considered as a topic for future research.

IV. CONCLUSION

This study proposes a device-independent shot detection method with low computational burden by taking the advantages of ORB descriptor and SSIM to achieve fast and accurate shot detection. The proposed method consists of three parts: the candidate segment preselection module, the cut transition detection module, and the gradual transition detection module. In the candidate segment preselection module, a lot of candidate segments with potential transitions are quickly obtained by comparing ORB features of twenty-frame intervals. Then, CT is detected in each candidate segment by comparing ORB and SSIM features of corresponding consecutive frames. Finally, GTs are detected in the candidate segments without CT based on the gradual transition model. The look-back mechanism is used in both CT detection and GT detection to improve the precision. The experimental result indicates that the proposed method can automatically detect the shots from test video sequences with a significant reduction in computational cost and achieve the prior performance among the compared methods by using ORM fused with SSIM.

In order to improve the performance of shot detection in unstable scenes of video frames, i.e. blurred frames and flickering light, the proposed method should be combined with other feature extraction algorithm, such as color information, shape features, and spatial location features etc. These issues are the subjects of our future work.

ACKNOWLEDGMENT

The authors would lke to thank all of the reviewers for their efforts and excellent suggestions in providing constructive comments, which help the authors to significantly enhance the quality of this paper. They also like to thank the editors for their valuable assistance during the review process.

REFERENCES

[1] J. Serres, D. Dray, F. Ruffier, and N. Franceschini, "A vision-based autopilot for a miniature air vehicle: Joint speed control and lateral obstacle avoidance," *Auton Robot*, vol. 25, nos. 1–2, pp. 103–122, Aug. 2008.

[2] F. Dufaux, P. Le Callet, R. Mantiuk, and M. Mrak, *High Dynamic Range Video: From Acquisition, to Display and Applications*. New York, NY, USA: Academic, 2016.

[3] D. Samaiya and K. K. Gupta, "Intelligent video surveillance for real time energy savings in smart buildings using HEVC compressed domain features," *Multimedia Tools Appl.*, vol. 77, no. 21, pp. 29059–29076, Nov. 2018.

[4] L. Hui-bin, W. Fei, C. Qiang, and P. Yong, "Recognition of individual object in focus people group based on deep learning," in *Proc. Int. Conf. Audio, Lang. Image Process. (ICALIP)*, Jul. 2016, pp. 615–619.

[5] Y. Gao, F. Villecco, M. Li, and W. Song, "Multi-scale permutation entropy based on improved lmd and hmm for rolling bearing diagnosis," *Entropy*, vol. 19, no. 4, p. 176, 2017.

[6] Y. Zhang, C. S. Nam, G. Zhou, J. Jin, X. Wang, and A. Cichocki, "Temporally constrained sparse group spatial patterns for motor imagery BCI," *IEEE Trans. Cybern.*, vol. 49, no. 9, pp. 3322–3332, Sep. 2019.

[7] H. Wang, Y. Zhang, N. R. Waytowich, D. J. Krusienski, G. Zhou, J. Jin, X. Wang, and A. Cichocki, "Discriminative feature extraction via multivariate linear regression for SSVEP–based BCI," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 24, no. 5, pp. 532–541, May 2016.

[8] B. Furht, E. Akar, and W. A. Andrews, *Digital Image Processing: Practical Approach*. Cham, Switzerland: Springer, 2018.

[9] Y. Jiao, Y. Zhang, Y. Wang, B. Wang, J. Jin, and X. Wang, "A novel multilayer correlation maximization model for improving CCA–based frequency recognition in SSVEP brain computer interface," *Int. J. Neural Syst.*, vol. 28, no. 04, May 2018, Art. no. 1750039.

[10] M. Liu, X. Wang, L. Nie, Q. Tian, B. Chen, and T.-S. Chua, "Cross-modal moment localization in videos," in *Proc. ACM Int. Conf. Multimedia*, 2018, pp. 843–851.

[11] M. Liu, X. Wang, L. Nie, X. He, B. Chen, and T.-S. Chua, "Attentive moment retrieval in videos," in *Proc. 41st Int. SIGIR Conf. Res. Develop. Inf. Retr.*, 2018, pp. 15–24.

[12] G. Gao and H. Ma, "To accelerate shot boundary detection by reducing detection region and scope," *Multimed Tools Appl*, vol. 71, no. 3, pp. 1749–1770, Aug. 2014.

[13] H.-B. Liu, T.-H. Tan, S.-C. Huang, Z.-J. Wang, and J. Zhou, "A novel shot detection approach using 8-neighbors and key-colors," in *Proc. 7th Int. Symp. Next Gener. Electron. (ISNE)*, May 2018, pp. 1–3.

[14] A. C. S. E. Santos and H. Pedrini, "Adaptive video shot detection improved by fusion of dissimilarity measures," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2016, pp. 002948–002953.

[15] A. F. Smeaton, P. Over, and A. R. Doherty, "Video shot boundary detection: Seven years of trecvid activity," *Comput. Vis. Image Understand.*, vol. 114, no. 4, pp. 411–418, 2010.

[16] W. Xu and L. Xu, "A novel shot detection algorithm based on clustering," in *Proc. 2nd Int. Conf. Educ. Technol. Comput.*, Jun. 2010, pp. V1-570–V1-572.

[17] Z.-M. Lu and Y. Shi, "Fast video shot boundary detection based on SVD and pattern matching," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 5136–5145, Dec. 2013.

[18] Z. Fang, F. Fei, Y. Fang, L. Shu, and W. Wan, "Abnormal event detection based on saliency information," *Int. J. Multimedia Ubiquitous Eng.*, vol. 10, no. 9, pp. 339–352, Oct. 2015.

[19] S. Domnic and G. G. L. Priya, "Walsh–hadamard transform Kernel-based feature vector for shot boundary detection," *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5187–5197, Dec. 2014.

[20] G. Mori, S. Belongie, and J. Malik, "Efficient shape matching using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 11, pp. 1832–1837, Nov. 2005.

[21] A. Hassanien, M. Elgharib, A. Selim, S.-H. Bae, M. Hefeeda, and W. Matusik, "Large-scale, fast and accurate shot boundary detection through spatio-temporal convolutional neural networks," 2017, *arXiv:1705.03281*. [Online]. Available: https://arxiv.org/abs/1705.03281

[22] M. Gygli, "Ridiculously fast shot boundary detection with fully convolutional neural networks," in *Proc. Int. Conf. Content-Based Multimedia Indexing (CBMI)*, Sep. 2018, pp. 1–4..

[23] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, 2012, pp. 2564–2571.

[24] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[25] I. D. Laboratory. (2019). *The Open Video Project*. [Online]. Available: https://open-video.org/index.php

[26] AImageLab. (2019). *Rai Dataset*. [Online]. Available: http://imagelab.ing.unimore.it/imagelab/researchActivity.asp?idActivity=19

[27] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 430–443.

[28] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 778–792.

[29] E. Karami, S. Prasad, and M. Shehata, "Image matching using sift, surf, brief and orb: Performance comparison for distorted images," 2017, *arXiv:1710.02726*. [Online]. Available: https://arxiv.org/abs/1710.02726

[30] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB–SLAM: A versatile and accurate monocular slam system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[31] R. Mur-Artal and J. D. Tardos, "ORB–SLAM2: An open–source slam system for monocular, stereo, and RGB–D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[32] D. M. Powers, "Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation," *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011.

[33] *Youtube*. (2019). [Online]. Available: https://www.youtube.com/watch?v=No-ZR7-X76s

[34] *Youku*. (2019). [Online]. Available: https://v.youku.com/v_show/id_XOTU0NzIzMTQw.html?spm=a2h0k.11417342.soresults.dtitle

[35] *Youku*. (2019). [Online]. Available: https://v.youku.com/v_show/id_XNjE2NDk4OTY=.html?spm=a2h0k.11417342.soresults.dtitle

HUIBIN LIU received the B.S. and M.S. degrees in computer science from Central South University, Changsha, in 2001 and 2005, respectively. She is currently pursuing the Ph.D. degree in electrical engineering with the National Taipei University of Technology, Taipei. She is also a Lecturer with the School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai. Her research interests include computer vision, machine learning, and artificial intelligence.

TAN-HSU TAN received the B.S. degree in electrical engineering from the National Taiwan Institute of Technology, in 1983, the M.S. degree in electrical engineering from National Tsing Hua University, in 1988, and the Ph.D. degree in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1998. Since 1988, he has been with the Department of Electrical Engineering, National Taipei University of Technology, Taipei, Taiwan, where he is currently a Professor. His research interests are wireless communications, telecare, machine learning, and optimization algorithms.

TIEN-YING KUO received the B.S. degree from National Taiwan University, Taiwan, in 1990, and the M.S. and Ph.D. degrees from the University of Southern California, Los Angeles, in 1995 and 1998, respectively, all in electrical engineering. In the summer of 1996, he worked as an Intern with the Department of Speech and Image Processing, AT&T Laboratories, Murray Hill, NJ, USA. In 1999, he was a Member of Technical Staff with the Digital Video Department, Sharp Laboratories of America, Huntington Beach, CA, USA. Since August 2000, he has been an Assistant Professor with the Department of Electrical Engineering, National Taipei University of Technology, Taipei, Taiwan, where he is currently an Associate Professor. He received the Best Paper Award from the IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), in 2008, and the Excellent Paper Award from Taiwan Academic Network Conference (TANET), in 2018.