AVCD-FRA: A novel solution to automatic video cut detection using fuzzy-rule-based approach

Roghayeh Dadashi, Hamidreza Rashidy Kanan

Department of Electrical, Computer and IT Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran

A R T I C L E   I N F O

Article history:
Received 11 August 2012
Accepted 18 March 2013
Available online 28 March 2013

Keywords:
Video content analysis
Video cut detection
Abrupt shot boundary detection
Hard cut detection
Fuzzy color histogram
Fuzzy-rule-base
Fuzzy logic

A B S T R A C T

Video shot boundary detection (SBD) is a fundamental step in automatic video content analysis toward video indexing, summarization and retrieval. Despite the beneficial previous works in the literature, reliable detection of video shots is still a challenging issue with many unsolved problems. In this paper, we focus on the problem of hard cut detection and propose an automatic algorithm in order to accurately determine abrupt transitions from video. We suggest a fuzzy rule-based scene cut identification approach in which a set of fuzzy rules are evaluated to detect cuts. The main advantage of the proposed method is that, we incorporate spatial and temporal features to describe video frames, and model cut situations according to temporal dependency of video frames as a set of fuzzy rules. Also, while existing cut detection algorithms are mainly threshold dependent; our method identifies cut transitions using a fuzzy logic which is more flexible. The proposed algorithm is evaluated on a variety of video sequences from different genres. Experimental results, in comparison with the most standard cut detection algorithms confirm our method is more robust to object and camera movements as well as illumination changes.

© 2013 Elsevier Inc. All rights reserved.

1. Introduction

Due to the expansion of the internet along with the rapid growth in multimedia technology and availability of digital storage devices with large capacity, vast video digital libraries are developed every day. Consequently, there exists a strong demand for techniques providing instant access and efficient indexing, browsing and retrieval of video data from these huge digital databases. The challenge of automatic processing of video data motivates researchers to advise efficient methodologies that organize video data and extract semantically meaningful information [1]. Segmenting video into elementary semantic units is a fundamental step in automatic video content analysis which is known as video shot boundary detection.

A video shot is a sequence of frames taken by one camera during a single continuous action in time and space [1,2]. The transition between shots is categorized into two types: cut and gradual transition. The cut transition is an abrupt change between two successive shots in which, the last frame of disappearing shot and the first frame of appearing shot are connected to each other immediately. Instead, in gradual transition (GT), two shots are connected to each other during a few frames using various editing effects.

Both cut and gradual shot transition detection is not a new task in automatic video content analysis and so far has been conducted great researches on this topic. Despite the recent advances, SBD on large scale video data is still a very difficult task [2]. In this paper, we focus on abrupt shot boundary detection from video sequences of different genres. Developing a cut detection algorithm that can efficiently identify cut transitions between adjacent shots which contain similar visual contents, and simultaneously is less sensitive to different types of content variations within a shot such as illumination changes and camera/object movements, is a challenging issue. Also, a useful cut detection algorithm should be able to perform well on videos from any genre.

It seems unattainable to establish a method which satisfies the above requirements concurrently. However, it is very beneficial to take into account a trade-off between the ability of detecting transition between similar shots and insensitivity to content variations within a shot. Towards this goal, we suggest a method which utilizes a fuzzy approach in order to detect cut transitions and intends to overcome the mentioned challenges in a unified approach, without any post processing step. The performance of the proposed cut detection algorithm is assessed on several challenging videos from different publicly available datasets. The obtained results, in contrast with the most standard shot detection algorithms, illustrate that our new method is more capable of accurately detecting cut transitions.

The rest of the paper is organized as follows: Section 2 provides a survey of some previous related works in the literature. Section 3...
discusses some existing problems and our solutions toward solving
them. In Section 4, the proposed method is introduced and its ele-
ments are studied in details. Experimental results are discussed in
Section 5 and Section 6 concludes the paper.

2. Related works

Till now, many efforts have been conducted in video cut detec-
tion area and many different methods have been introduced. Exis-
ting algorithms identify cut transitions based on this fact that visual
contents of two successive frames belonging to distinct shots are
ideally very different while sequential frames within a shot have
very similar visual contents. As a result, each cut detection tech-
nique is a kind of classification procedure composed of two stages:
calculation of visual content dissimilarities, also known as feature
extraction phase, and decision-making. Various algorithms for iden-
tifying cut transitions adopt different approaches in these two
stages. In the following, we will study some existing works con-
cerning on video shot boundary detection especially hard cut
detection approaches.

Pixel-based technique is the simplest method to detect cut tran-
sitions in a video sequence. The algorithm relies on pair-wise pixel
intensity comparison of two successive frames. A cut is found if
the number of pixels that vary more than a threshold, exceeds a second
threshold [3]. Since this method reflects any details of frames, it is
susceptible to noise and camera/object motions [3–5]. Zhang et al.
[6] improved the performance of the pixel-based approach by low
pass filtering before the comparison. Cernekova et al. [7] located
cut transitions using calculating the mutual information via the
variations of corresponding pixels between sequential frames.
They demonstrated that in contrast with pixel differences, the
information theory measure captures the frame-to-frame informa-
tion more compactly and so it provides better results.

Gray level or color histograms are a prevalent alternative to pix-
el-based approaches [4]. Histogram-based techniques are easy to
calculate and since they do not carry spatial information of pixels,
they are more robust to object/camera motions. For these reasons
they are the most popular approaches to detect cut transitions.
Histogram difference (HD) is the most straightforward way to
the histogram-based cut detection methods. Whenever sum of
the absolute bin-wise difference between histogram of two consec-
tutive frames surpasses a predefined threshold, a cut is recognized.
In such an approach, selecting an appropriate threshold is a major
issue. Dugad et al. [8] improved threshold depending HD method
using a two step strategy. Former, they compared the intensity his-
togram difference (IHD) of successive frames against a large local
threshold. If this value exceeds the threshold, a shot is found else,
in the next step, for the IHD value more than a small local thresh-
old, likelihood ratio is computed between the two frames. The like-
lihood ratios more than a fixed threshold cause to find remaining
change. Considering the drawbacks of using a static threshold,
several adaptive thresholding-based cut detection methods were
introduced [9–11]. Yu and Sinhath [9] employed an entropic tech-
nique to attain a specific threshold for any video sequence auto-
matically. They compared gray scale histogram differences of
consecutive frames against this threshold for identifying candidate
scene cuts. Yusoff et al. [10] presented some new methods for
obtaining the threshold locally using a sliding window. They
experimentally demonstrated that cut detection using their strat-
egies outperform the adaptive method introduced in [8] and other
non-adaptive methods. However, selecting optimum window size
is a key issue and affects detection performance.

There have been some efforts toward characterizing shot tran-
sitions using a learning based methodology instead of simple
thresholding. Chasnatis et al. [12] and Ling et al. [13] trained a sup-
port vector machine (SVM) classifier to locate shot transitions.
Adopting machine learning methods encounters with two major
problems [4]. Building appropriate features for the classifier is
the first one. Chasnatis et al. [12] constructed a feature vector with
120 elements for each video frame using the RGB histogram
squared distance values in a sliding window. Ling et al. [13]
combined a combination of intensity pixel differences, edge histo-
gram differences in horizontal and vertical directions, hue and sat-
uration color histogram differences in HSV color space to build a
five-dimensional feature vector for each video frame as the input
vector to the SVM. The next key problem is that the classifier needs
to train with a suitable video set containing of moderately bal-
anced positive (normal frames) and negative (transitional frames)
examples [4].

Comparing video frames based on histogram features has the
disadvantage of being sensitive to illumination changes. Some of
the researchers intended to cope with this problem by ignoring
features that carry luminance information. It can eliminate some
problems due to brightness changes but also loses the luminance
information which could be important in determining the variation
of visual contents in video sequence [4]. For example, Lefevre and
Vincent [11] employed color histogram of the hue and saturation
in HSL color space to avoid the disturbance of the lighting varia-
tions. They computed frame dissimilarities through a measure de-

erived in a hue-saturation color space, and determined shot
boundaries using an adaptive threshold. Zhou and Zhang [14] con-
sidered the normalized RGB color model, which is invariant to light
intensity changes, and utilized only the histogram of color compo-
nent r with 256 bins. They reduced the dimensions of the histo-
gram by independent component analysis (ICA) technique and
subsequently represented each video frame by only a two-dimen-
sional feature vector. Afterwards, they applied an iterative cluster-
ing algorithm to detect both cuts and gradual transitions.

Zabih et al. [15] introduced an edge change ratio (ECR) method
in order to intensity invariant shot detection. Regarding to experi-
ments, ECR-based hard cut detection does not outperform histo-
gram-based methods and needs much more computations [16].
Also, edge based method suffers from camera zooming and scaling,
noise, and sudden appearance/disappearance of an object. How-
ever it has application in reducing false detections due to sudden
change in illumination [1]. Das et al. [1], in a post processing stage,
added edge based technique to eliminate false positives caused
by abrupt brightness change. To detect cut boundaries, they
exploited fuzzy low level features, namely fuzzy histogram and
fuzzy co-occurrence matrix, and appraised a set of fuzzy rules,
modeled by interval type-2 sets. Qing et al. [17] introduced an illu-
mination invariant metric to measure the dissimilarity of video
frames. Specially, they applied this metric to candidate cuts in or-
der to get rid of false detections caused by intensity change. They
empirically illustrated that their metric could eliminate some of
the false positives successfully without removing any true cuts.

Recently, Küçüktunç et al. [18] presented a novel fuzzy color
histogram (FCH) and used it to detect shot boundaries for video
copy detection task. FCH which is obtained from video frames in
CIELAB color space is more accurate than other color histograms
and more stable against illumination changes [18,19]. Considering
these advantages, we also adopt FCH to elicit visual features from
video frames.

In recent years, some of the researchers draw out more robust
and complex features of video frames. For instance, Liu et al. [20]
computed the visual content dissimilarities in digital video using
scale invariant feature transform (SIFT). Barbu [21] computed a tri-
dimensional feature vector for video frames using two-dimen-
sional Gabor filtering. He also provided a region-growing based
classification approach to categorize the distances (sum of absolute
differences) between feature vectors of consecutive frames.
Obtained results revealed that these methods perform well, however they are sophisticated and computationally very expensive.

Recently, Amiri and Fathy [2,22] put forward linear algebra based techniques in order to provide a satisfactory detection performance. In [2], they used the properties of QR-decomposition to extract a probability function that indicates the probability of belonging video frames to shot transitions. They also defined a distance function in Eigen space using the properties of generalized eigenvalue decomposition for comparing video frames [22]. They located abrupt and gradual shot boundaries by analyzing the behavior of these functions along the time domain. Despite the high performance, these techniques fail in strong flashlights and they are very time consuming due to extensive matrix operations.

According to the above discussion, we can see many efforts have been devoted to detect video shot boundaries, and a lot success has been obtained. However, in real applications, object motions, camera operations, and illumination variations are often mistaken as shot boundaries. The main reason is that conventional methods only take into account difference of successive frames in order to detect cut transitions. Overcoming the aforementioned problems requires that visual content variations along the time domain to be considered. Consequently, in the proposed method, cut transitions are modeled as a set of fuzzy rules based on visual content variations around each video frame. These fuzzy rules are sufficiently discriminant between cut transitions and content variations within a shot. More precisely, to represent visual contents of video sequences, we extract localized fuzzy color histogram (FCH) [18] for each frame. Incorporating spatial and temporal information, a four-dimensional feature vector is created for each video frame based on FCH distances in a sliding temporal window. The feature vectors are used as inputs of a fuzzy system to evaluate the set of fuzzy rules. The evaluation result determines if a frame belongs to a ‘cut transition’ or an ‘intra shot’ frame. The main advantage of the proposed method is its satisfactory robustness to camera flash lights, object motions, and camera panning/tilting. Also the detection process is automatic and there is no use of any threshold.

3. Problems and motivations

In this section, we discuss the major problems in the literature and describe our approach toward solving them efficiently.

- Existing cut transition detection methods are mostly threshold dependent which are susceptible to the content of the video being processed and do not generalize well for various video types [12]. Such techniques which detect cut transitions using a “crisp” rule are coarse. In contrast, we adapt a fuzzy method which is more flexible and there is no need to any threshold adjusting.

- Representing visual content of video frames using appropriate features is a significant step in shot boundary detection approaches and impresses the overall performance of the algorithm. The visual features should remain constant through a distinct shot and reflect content variations during a shot transition. Also, the features must be easy to compute and insensitive to variations of intra shot visual parameters such as rotation as much as possible. Several empirical assessments have revealed that simple color histogram is able to attain a satisfactory result while some complicated features such as edge cannot outperform the simple feature [4]. Since fuzzy color histogram (FCH) is more accurate than a conventional color histogram [18], we have calculated fuzzy color histogram of each video frame. Also, the size of the area from which visual features are exploited, affects the detection efficiency. A small area (single pixel) decreases stability with respect to motion, while a large area (whole frame) tends to lose transitions between similar shots. To achieve a satisfying detection performance, we consider a trade-off between them, and consequently segment each frame into non-overlapping equal-sized blocks and extract fuzzy color histogram of each block. This approach is sufficiently discriminant for shot transitions and adequately robust to small object and camera movements.

- Usually, existing cut transition detection methods are unable to distinguish transitions between the shots which contain similar visual contents. Detection of transitions between similar shots often increases the number of false positives due to the content variations during a single shot. Also, sudden illumination changes such as flashlights during a single shot are often confused with shot boundaries especially in color histogram based methods in which luminance is a basic component. As mentioned in previous section, several intensity invariant features and similarity metrics devised to ameliorate this drawback. However, these solutions only moderate the problem and do not resolve it completely; they also might lead to missed detections due to lack of luminance information.

To overcome these major problems, considering spatial and temporal dependency of video frames can be very useful. As a result, for each video frame, we construct a feature vector which describes the variations of frames in a temporal window surrounding the desired frame, using FCH distances. This feature vector efficiently distinguishes cut transitions from intra shot content variations. We also formulate cut transitions via temporal relation of video frames as a set of fuzzy rules. The feature vectors are employed to evaluate these fuzzy rules to determine true cut transitions.

4. The proposed AVCD-FRA method

A shot boundary detection task is a kind of classification problem along the time domain. During this classification process, it is required to construct a feature vector for each video frame based on the statistics of visual attributes of the frame and its some adjacent frames. These feature vectors are utilized as the inputs of a shot detection classifier. In this paper, we extract block-based fuzzy color histogram from video frames and create a feature vector for each frame using the dissimilarity values between the FCHs of frames in a temporal window surrounding the frame of the interest. These feature vectors are used as inputs to a fuzzy inference system to evaluate a set of fuzzy rules. The evaluation result determines where a cut transition occurs.

The block diagram of the proposed method is demonstrated in Fig. 1. As shown, the framework consists of three phases, which are block-based fuzzy color histogram extraction, feature vector
construction and classification using a fuzzy inference system. In the following, we will study these sections in details.

4.1. Block-based fuzzy color histogram

As mentioned before, in order to develop a cut detection method which is invariant to motions as well as being adequately discriminant for shot boundary detection, we split each frame into non-overlapping equal-sized blocks and extract a normalized histogram of each block. Among various color histograms, we have chosen fuzzy color histogram because it is less sensitive and more robust to illumination changes and quantization errors [18]. This histogram consists of 15 bins such that each bin corresponds to a special color. FCH is generated using a fuzzy system which its input

![Feature vector for a normal frame.](image1)

![Feature vector for a hard cut.](image2)

![Feature vector for a hard cut with similar visual content.](image3)

![Feature vector for a frame containing a flashlight.](image4)

![Feature vector for a frame within a gradual transition.](image5)

![Feature vector for a frame at the beginning of a GT.](image6)

Fig. 2. The feature vector values for several frames in various situations (each feature vector corresponds to the frame at the top).
and output are an image pixel in CIELAB color space and a crisp value correspond to a bin of the histogram, respectively. More details have been provided in [18,19]. We normalize the histogram by dividing the values of the histogram by the total number of pixels.

4.2. Feature vector construction

Constructing the feature vectors for video frames is a crucial step in cut detection problem. Selected features should adequately characterize the category of the frame as a cut transition or intra shot frame. For each video frame, we create a feature vector composed of four elements. These elements are the dissimilarity/distance values calculated in a temporal window surrounding the frame of interest. More precisely, for each frame \( I_\text{t} \), the feature vector \( F_{\text{t}} \) is formed as follows:

\[
F_{\text{t}} = \begin{cases} 
D(I_\text{t}, I_{\text{t}-1}) , & \text{FF:First Feature} \\
D(I_{\text{t}-1}, I_{\text{t}+1}) , & \text{SF:Second Feature} \\
D(I_{\text{t}-1}, I_{\text{t}+2}) , & \text{TF:Third Feature} \\
\min_{1 \leq i \leq 3} \left( D(I_{\text{t}-1}, I_{\text{t}+j}) \right) , & \text{LF:Last Feature}
\end{cases}
\]

In Eq. (1), function \( D \) computes the mean of the dissimilarity/distance between FCHs of two desired frames as follows:

\[
D(I_\text{t}, I_j) = \frac{1}{n} \sum_{k=1}^{n} d(FCH_i^k, FCH_j^k)
\]

(2)

where \( n \) is the number of blocks and \( d(FCH_i^k, FCH_j^k) \) calculates the Euclidean distance between FCHs of block \( k \) of frames \( I_\text{t} \) and \( I_j \):

\[
d(FCH_i^k, FCH_j^k) = \sqrt{\sum_{p=1}^{15} (FCH_i^k(p) - FCH_j^k(p))^2}
\]

(3)

We use first and second features to detect cut transitions regard to the fact that frame-to-frame dissimilarity values within a shot are small and in cut transition positions increase dramatically. Although, these features are sufficient to characterize hard cuts, but also lead to false alarms due to content variations within a shot (such as object and camera movements or computer generated features). The third feature, \( D(I_{\text{t}-1}, I_{\text{t}+2}) \), helps us to distinguish discontinuities due to hard cuts from those which are due to intra shot content variations (also gradual transitions). If there exist an abrupt transition between \( I_\text{t} \) and \( I_{\text{t}+1} \), then the amount of \( D(I_{\text{t}-1}, I_{\text{t}+2}) \) should be a small value since \( I_{\text{t}-1} \) and \( I_{\text{t}+2} \) are adjacent and belong to a same shot; whereas, noticeable value of \( D(I_{\text{t}-1}, I_{\text{t}+2}) \) means the discontinuity between \( I_\text{t} \) and \( I_{\text{t}+1} \) is due to the mentioned reasons. The last feature can be useful to eliminate false alarms due to significant illumination and content variations especially camera flash lights during a shot. Using this feature, we investigate the dissimilarity values between two groups of frames, \( (I_{\text{t}-3}, I_{\text{t}-2}, I_{\text{t}-1}, I_\text{t}) \) and \( (I_{\text{t}-1}, I_{\text{t}+2}, I_{\text{t}+3}, I_{\text{t}+4}) \) aimed to discover the minimum dissimilarity. If there actually exist an abrupt transition between \( I_\text{t} \) and \( I_{\text{t}+1} \), then this value should be a high amount. Fig. 2 demonstrates feature vector values for several frames in various situations.

4.3. Frame classification using fuzzy-rule-based system

After the feature definition, a suitable classifying method has to be utilized in order to classify each video frame into ‘hard cut’ and ‘intra shot’ frame categories. To achieve this goal, we use a fuzzy
inference system since the cut detection problem could be modeled satisfactorily as a set of fuzzy rules. Fig. 3 demonstrates the basic structure of the proposed fuzzy system. The four feature values, generated for each frame, are used as inputs of the fuzzy system. Output of the system is a crisp value which indicates the category of the frame that belongs to. As shown in Fig. 3, the system is composed of four major modules: fuzzifier, cut detection fuzzy rule base, fuzzy inference engine, and defuzzifier. In the following, we will explain our methodologies used in each of these modules.

- **Fuzzifier**: The fuzzification of the inputs is performed according to fuzzy membership functions shown in Fig. 4. We have determined the membership functions empirically.

- **Cut detection fuzzy rule base**: Based on theoretical analysis in Section 4.2, we have formulated cut transitions as a set of fuzzy rules as illustrated in Fig. 5. Cut transitions are detected by evaluation of these rules.

- **Fuzzy inference engine**: A Mamdani style fuzzy inference engine evaluates the mentioned fuzzy rules in which implication method and AND operator for evaluation of rules are both set

<table>
<thead>
<tr>
<th>Rule</th>
<th>Conditions</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>(FF is smaller) and (SF is higher) and (TF is small) and (LF is higher) then (output is cut).</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>(FF is smaller) and (SF is higher) and (TF is small) and (LF is lower) then (output is cut).</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>(FF is smaller) and (SF is high) and (TF is small) and (LF is higher) then (output is cut).</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>(FF is smaller) and (SF is high) and (TF is high) and (LF is higher) then (output is cut).</td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>(FF is smaller) and (SF is mid) and (TF is small) and (LF is higher) then (output is cut).</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>(FF is smaller) and (SF is mid) and (TF is high) and (LF is higher) then (output is cut).</td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>(FF is small) and (SF is higher) and (TF is small) and (LF is higher) then (output is cut).</td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>(FF is small) and (SF is high) and (TF is small) and (LF is higher) then (output is cut).</td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td>(FF is small) and (SF is high) and (TF is high) and (LF is higher) then (output is cut).</td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td>(FF is mid) and (SF is high) and (TF is small) and (LF is higher) then (output is cut).</td>
<td></td>
</tr>
<tr>
<td>11.</td>
<td>(FF is mid) and (SF is high) and (TF is high) and (LF is higher) then (output is cut).</td>
<td></td>
</tr>
<tr>
<td>12.</td>
<td>(FF is high) or (SF is small) or (TF is higher) or (LF is high) then (output is intra-shot).</td>
<td></td>
</tr>
<tr>
<td>13.</td>
<td>(FF is high) or (SF is small) or (TF is higher) or (LF is high) then (output is intra-shot).</td>
<td></td>
</tr>
</tbody>
</table>

**Step 1**

for each frame $I_i$ in the video sequence do

1. segment $I_i$ into $k \times k$ non-overlapping blocks
2. for each block in $I_i$, $I_i^b$ do
   - (2-1) $FCH^b_i \leftarrow$ compute normalized Fuzzy Color Histogram of $I_i^b$

**Step 2**

for each frame $I_i$ in $\{I_4, I_5, ..., I_{n-4}\}$ do

1. compute the $F_i$ using Equations (1-3)
2. Decision Value $\leftarrow$ run fuzzy system described in section (4.3) on $F_i$ and return its crisp output
3. if (Decision Value $> 0$) then
   - $I_i$ is a “hard cut”
   else
   - $I_i$ is an “intra shot frame”

**Fig. 6.** Fuzzy rules for cut transition detection.

**Fig. 7.** The procedure of the proposed video cut detection method.
to MIN. Obviously, the output of the system has two membership functions corresponding to ‘hard cut’ and ‘intra shot’ frame categories as shown in Fig. 6.

Aggregation process for combination fuzzy sets, representing the output of each rule, into a single fuzzy set is accomplished by MAX method.

- **Difuuzifier**: The fuzzy set obtained from aggregation process, is defuzzified using centroid method to produce a crisp decision value. The frame is classified as a ‘hard cut’ when this value indicates a positive result.

Fig. 7 summarizes the procedure of the proposed video cut detection method.

### 5. Experimental results

The performance of the proposed abrupt shot boundary detection technique was evaluated on several various videos. Experiments were carried out on totally 15 videos from different categories containing news, cartoon, sport, movie, commercial, and documentary. The video data were taken from famous video datasets such as ‘TRECVID 2006’ [23], ‘Open Video Project’ [24], ‘MOCA project’ [25], and ‘Video Segmentation Project in Carleton University’ [26] which all of them are publicly available on the Internet. The details of the test videos have been presented in Table 1. Besides cut transitions, the video sequences especially ‘N1’ and ‘N2’ include a large number of gradual transitions. Therefore, we could test the ability of the proposed algorithm in distinguishing between hard cut and gradual transitions, too. The test video sequences include variety of video effects, camera and object motions, and sharp illumination changes due to strong camera flashes. To evaluate the performance of the proposed cut detection method, we utilized the following metrics [12,21]:

- **Precision**, as a measure of quality which is the percentage of correct detections:
  \[
  \text{Precision} = \frac{\text{Number of correct detections}}{\text{Number of all detections}} \times 100
  \]

- **Recall**, as a measure of quantity, which is the percentage of detected true cuts:
  \[
  \text{Recall} = \frac{\text{Number of correct detections}}{\text{Number of true cuts}} \times 100
  \]

- **F1 score**, which provides a single measurement using combination of precision and recall:
  \[
  F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \times 100
  \]

In the experiments, we assimilated the size of all video frames by resizing them to 180×240 pixels. Also, to detect cut transitions using the mentioned algorithm which is shown in Fig. 7, we divided each video frame to 5×5 equal-sized blocks and extracted fuzzy color histogram of each block. Table 2 displays the obtained results in terms of precision, recall and F1 for each test video separately.

As it can be seen from Table 2, the proposed method achieves 100% recall with only 66.7% in precision for video N3 and S2. This amount of precision for video N3, which contains lots of quick camera and object movements, is not surprising. The proposed method produces only two false positives for this video sequence; however, since there are just four detected true cuts, it is evident the precision value decreases due to these false alarms dramatically. Fig. 8 shows the two false detections produced by the algorithm for video N3.

Video S2 includes two cuts and one gradual transition (push-type transition) consisting of one transitional frame. This gradual transition, shown in Fig. 9, causes a false positive by the algorithm which leads a significant reduction in precision value.

### Table 1

**Description of the experimental video sets.**

<table>
<thead>
<tr>
<th>Category</th>
<th>ID</th>
<th>Video Name</th>
<th>Frames</th>
<th>Cuts</th>
<th>GT</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>News</td>
<td>N1</td>
<td>20051114_091300_NTDTV_FOCUSINT_CHN.mpg</td>
<td>31,138</td>
<td>121</td>
<td>78</td>
<td>TRECVID 2006</td>
</tr>
<tr>
<td></td>
<td>N2</td>
<td>20051227_125800_CNNT져FROM_ENG.mpg</td>
<td>9772</td>
<td>44</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td></td>
<td>N3</td>
<td>News.mpeg</td>
<td>479</td>
<td>4</td>
<td>0</td>
<td>Carleton University</td>
</tr>
<tr>
<td>Cartoon</td>
<td>C1</td>
<td>CARTOON.mpg</td>
<td>500</td>
<td>11</td>
<td>1</td>
<td>MOCA project</td>
</tr>
<tr>
<td></td>
<td>C2</td>
<td>Lisa.mpeg</td>
<td>650</td>
<td>7</td>
<td>1</td>
<td>Carleton University</td>
</tr>
<tr>
<td>Sport</td>
<td>S1</td>
<td>SOCCER MPEG</td>
<td>500</td>
<td>2</td>
<td>0</td>
<td>MOCA project</td>
</tr>
<tr>
<td></td>
<td>S2</td>
<td>TENNIS.mpeg</td>
<td>500</td>
<td>2</td>
<td>1</td>
<td>Carleton University</td>
</tr>
<tr>
<td></td>
<td>S3</td>
<td>SOCCER_GAOL.mpg</td>
<td>750</td>
<td>1</td>
<td>2</td>
<td>Internet [27]</td>
</tr>
<tr>
<td>Movie</td>
<td>M1</td>
<td>VideoAbstract.mpeg</td>
<td>5132</td>
<td>40</td>
<td>0</td>
<td>Carleton University</td>
</tr>
<tr>
<td></td>
<td>M2</td>
<td>Highlander_2.mpeg</td>
<td>536</td>
<td>26</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>M3</td>
<td>D.mpeg</td>
<td>2632</td>
<td>34</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Commercial</td>
<td>Co1</td>
<td>Commercial2.mpeg</td>
<td>236</td>
<td>0</td>
<td>3</td>
<td>Carleton University</td>
</tr>
<tr>
<td></td>
<td>Co2</td>
<td>COMMERCIAL.mpeg</td>
<td>500</td>
<td>18</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Documentary</td>
<td>D1</td>
<td>Indi001.mpg</td>
<td>1687</td>
<td>15</td>
<td>1</td>
<td>Open Video Project</td>
</tr>
<tr>
<td></td>
<td>D2</td>
<td>Bor03_001.mpeg</td>
<td>3183</td>
<td>18</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td>58,195</td>
<td>343</td>
<td>120</td>
<td></td>
</tr>
</tbody>
</table>

### Table 2

**Experimental results of the proposed cut detection algorithm.**

<table>
<thead>
<tr>
<th>Video ID</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>86</td>
<td>96.7</td>
<td>91</td>
</tr>
<tr>
<td>N2</td>
<td>90.7</td>
<td>88.6</td>
<td>89.6</td>
</tr>
<tr>
<td>C1</td>
<td>91.7</td>
<td>100</td>
<td>95.7</td>
</tr>
<tr>
<td>C2</td>
<td>87.5</td>
<td>100</td>
<td>93.3</td>
</tr>
<tr>
<td>S1</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>S2</td>
<td>66.7</td>
<td>100</td>
<td>80</td>
</tr>
<tr>
<td>S3</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>M1</td>
<td>100</td>
<td>92.5</td>
<td>96.1</td>
</tr>
<tr>
<td>M2</td>
<td>85.7</td>
<td>92.3</td>
<td>88.9</td>
</tr>
<tr>
<td>M3</td>
<td>97.1</td>
<td>100</td>
<td>98.5</td>
</tr>
<tr>
<td>Co1</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Co2</td>
<td>100</td>
<td>88.9</td>
<td>94.1</td>
</tr>
<tr>
<td>D1</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>D2</td>
<td>94.7</td>
<td>100</td>
<td>97.3</td>
</tr>
<tr>
<td>Total</td>
<td>90.6</td>
<td>95.3</td>
<td>92.9</td>
</tr>
</tbody>
</table>
In general, the algorithm satisfactorily conquers content variations due to illumination changes. For instance, video N1 contains lots of strong camera flashlights which none of them are mistakenly classified as cuts by the algorithm. False detections are mainly due to some gradual transitions in which content variations are impressive and similar to abrupt transitions. Also, missed detections mostly result in transition between two successive shots in which the frame pairs at shot boundaries contain very similar visual attributes. Fig. 10 shows some examples of missed detections. Fig. 11 displays examples of non-cut unexpected content variations that the algorithm successfully overcomes. Totally, the proposed cut detection algorithm obtains 95.3% recall with 90.6% in precision. It is clear our method achieves a good trade-off between precision and recall with a high amount in $F_1$ measure.

To investigate the performance of the proposed algorithm over threshold dependent methods, we compared it with the following techniques:

- Fuzzy color histogram-based video segmentation [18]

In this approach, global FCH is extracted from video frames and then, for detecting cut transitions, color histogram dissimilarity of each frame $I_t$ is calculated as follows:

$$d(I_t) = \frac{1}{2}(D(I_t, I_{t-2}) + D(I_{t+1}, I_{t-1}))$$

where $D(I_n, I_m)$ is the Euclidean distance of FCH of frames $I_n$ and $I_m$. An abrupt shot transition between $I_t$ and $I_{t+1}$ is detected when $d(I_t)$ exceeds a certain global threshold, $\theta_c = 0.15$. However, best results in our experiments were achieved with $\theta_c = 0.17$.

- Information theory-based shot cut detection [7]

In this approach, cut transitions are detected using mutual information (MI) between two successive frames. It is claimed that hard cut boundaries are located efficiently even in the presence of fast object and camera movements. Generally, a small MI value between two adjacent frames indicates the existence of a cut between them. To detect hard cuts, local MI mean values on a temporal window $W$ of size $N_w$, centered on time instance $t_c$, are computed as follows:

$$I_{t_c} = \frac{1}{N_w} \sum_{t \in W} I_{t+c}$$

where $I_{t+c}$ is the MI value between frame $f_t$ and $f_{t+1}$. If the quantity $I_{t_c} / I_{t+c-1}$ surpasses a prefixed threshold $\theta_c$. In some cases, for example flash lights or a special type of hard cuts, the MI values show double peaks. This drawback is enhanced by a pre-processing step; all double peaks are checked for the possibility of corresponding to a hard cut, via comparing the previous and the next frames. Where, the MI value
is still small, it is modified to become a single peak [7]. In our experiments, similar to [7], we used the size of temporal window \( N_w = 3 \) and with a little difference, the best global threshold for our videos was obtained as \( h_c = 3.2 \).

- CutDet in MOCA project [16]

  It is one of the most reliable variants of histogram-based detection algorithms [16,7]. In this approach, hard cuts can be detected as single peaks in the time series of the differences between RGB color histograms of adjacent frames. Thus, a cut change is detected between \( l_{t-1} \) and \( l_t \) when \( CHD_l(\text{color histogram difference between } l_{t-1} \text{ and } l_t) \) exceeds a predefined global threshold, \( \theta_c \). In order to cope with a very particular type of hard cut which consists of one transitional frame, in a pre-processing stage double peaks are modified into single peaks at the higher \( CHD_l \) [16]. In our experiments, best performance using this method were produced with \( \theta_c = 0.4 \).

We applied the three shot boundary detection algorithms on video sequences listed in Table 1. Table 3 illustrates the performance of the proposed algorithm compared with the comparative methods (Refs. [18,7] and CutDet) among all the video genres inside the test data set. From the presented results in Table 3, it can be observed that the proposed method performs for news, cartoon, and documentary categories better than the CutDet method and maintains the same performance for sport video class.

In commercial and movie categories, the obtained precision by the CutDet is higher than the proposed algorithm but the recall value especially in commercial category is very low. However, in terms of \( F_1 \) measure, which is a combination of precision and recall, our method outperforms it for both movie and commercial videos.

In contrast with Ref. [18], our method achieves more reliable performance for news, cartoon, sport, commercial, and documentary categories. For movie video genre, the obtained precision by Ref. [18] is slightly higher than our technique while its recall value is significantly lower than ours. Again, using \( F_1 \) metric, the proposed algorithm outperforms for movie video group as well.

Compared with Ref. [7], the AVCD-FRA method outperforms in terms of precision and recall for all video categories except cartoon video set. As shown in Table 3, for this video type, obtained precision by the proposed method and Ref. [7] is 90–100% whereas the recall value is 100–83%. Therefore, as the \( F_1 \) quantity verifies, it is evident our algorithm operates more reliable for cartoon video type too.
In summary, the proposed AVCD-FRA method acts 11.1%, 8.8% and 16.1% more accurate than Refs. [18,7] and CutDet, respectively. Also, using recall value, it outperforms Refs. [18,7] and CutDet, 14.9%, 9.9% and 10.2%, respectively. To provide an overall comparison, we summarized the experimental results using $F_1$ measure as graph in Fig. 12. Totally, the obtained results indicate that the proposed cut detection algorithm is more reliable than the three other algorithms.

6. Conclusion

In this paper, we proposed an automatic abrupt shot boundary detection technique which detects cut transitions in three steps: (1) visual attributes are extracted from video frames by calculating localized fuzzy color histogram of each frame; (2) for each video frame, a feature vector is constructed using fuzzy color histogram distances in a temporal window; and (3) video frames are classified
as ‘cut’ and ‘inter shot frame’ using a fuzzy inference system in which the input is the feature vectors constructed in previous step. The main advantage of the proposed method is that the detection process is automatic and no threshold or other parameter is used. As shown by the experiments, the proposed method performs much better than the three representative threshold dependent algorithms. In summary, 95.3% recall with 90.6% in precision obtained on the test video set from various video categories whereas the best precision and recall produced by the three other algorithms were 81.8% and 85.4%, respectively. Generally, the proposed method is able to deal with camera flashlights. False detections are rarely due to quick camera movements and computer generated features, but are mainly due to gradual transitions in the videos. Therefore, in the feature work, we will try to develop the proposed cut detection method for various gradual transition types.

References