EFFICIENT VIDEO INDEXING FOR FAST-MOTION VIDEO

Min-Ho Park and Rae-Hong Park

Department of Electronic Engineering, School of Engineering, Sogang University
35 Baekbeom-ro (sinsu-dong), Mapo-gu, Seoul 121-742, Korea

ABSTRACT

Due to advances in recent multimedia technologies, various digital video contents become available from different multimedia sources. Efficient management, storage, coding, and indexing of video are required because video contains lots of visual information and requires a large amount of memory. This paper proposes an efficient video indexing method for video with rapid motion or fast illumination change, in which motion information and feature points of specific objects are used. For accurate shot boundary detection, we make use of two steps: block matching algorithm to obtain accurate motion information and modified displaced frame difference to compensate for the error in existing methods. We also propose an object matching algorithm based on the scale invariant feature transform, which uses feature points to group shots semantically. Computer simulation with five fast-motion video shows the effectiveness of the proposed video indexing method.

KEYWORDS

Scene Change Detection, Shot Boundary Detection, Key Frame, Block Matching Algorithm, Displaced Frame Difference, Scale Invariant Feature Transform, Video Indexing.

1. INTRODUCTION

Recently, computing speed and memory capacity have increased greatly. With personal computer or mobile multimedia devices such as smart phone, personal digital assistant, notebook, and so on, one can connect to the Internet and surf to get, copy, browse, and transmit digital information easily. Demand for multimedia information such as text, document, images, music, and video has also increased according to the rapid advance in multimedia consumer products or devices.

Mostly text-based search services have been provided rather than multimedia information search services such as music video that requires huge memory and a high computational load. Digitized multimedia information is searched at the sites which are connected through the Internet and used for various multimedia applications. Different from text-based information, it is relatively difficult to generate keywords for search of multimedia information. Also automatic interpretation of image or video is a big challenge. Text-based information can be analyzed and understood by simply recognizing characters themselves, while it is a challenging problem to develop a general algorithm that can analyze and understand nontext-based information such as video, photos, and music that are produced under various conditions. The text-based search services have a limit on the amount of information because of the rapid spread of multimedia consumer products or devices and the Internet. The huge amount of digital multimedia contents available today in digital libraries and on the Internet requires effective tools for accessing and browsing visual data

DOI : 10.5121/ijcga.2014.4204

information.
Video is composed of many scenes each of which involves one story unit. This story unit consists of many shots that contain visual content elements. We can understand the story by recognizing these elements. Thus, content elements are very important to summarize or retrieve video sequences. Shot boundary detection (SBD) is an essential step to efficient video search and browsing, and thus a large number of SBD methods have been presented. Most SBD methods are reasonably effective for specific types of video they are developed for, however it is noted that a single SBD method using a simple low-level image feature cannot work for all types of scene boundaries, especially for general video sequences as human viewers can do. Most of existing SBD methods has drawbacks in processing video that contains rapid illumination changes or fast moving objects and background [12].

As a solution to these problems and for efficient search, storage, and management of videos, SBD and video content based indexing methods that group similar shots have been developed. SBD finds shot boundaries of scene changes and splits a video into a number of video segments. For SBD, features extracted from a given video include intensity/color difference, motion vector (MV), texture, and so on. For performance improvement, more than two features are combined to give reliable results that are robust to illumination change and abrupt or complex motion. Video indexing is performed by computing similarities between input video and representative key frames selected among shots.

For efficient video indexing of fast-motion video, this paper uses our previous work on SBD [12, 15]. Our previous SBD method uses the modified intensity difference and MV features for fast-motion video containing motion blur. The similarity between key frames extracted from detected shots is computed using the scale invariant feature transform (SIFT) that is invariant to size and illumination change. By using the computed similarity as a measure to group similar shots, a video is decomposed into a number of meaningful scenes, in which a number of frames including the key frame are used for robustness to motion blur or rotation of objects in a scene.

The main contributions of the paper are: (i) object based shot grouping using the SIFT descriptor, and (ii) shot change detection and video indexing for fast-motion video using motion information. The rest of the paper is structured as follows. Section 2 reviews on SBD and video indexing. Section 3 proposes a video indexing method using our previous SBD methods for fast-motion video. Section 4 gives experimental results of video indexing and discussions. Finally, Section 5 draws conclusion of the paper.

2. RELATED WORK: SBD AND VIDEO INDEXING

Compared with other multimedia information, video requires huge memory storage and contains a vast amount of information as well. By character recognition, text-based information can be easily and heuristically retrieved and understood. However, video analysis is much more complex than text analysis and there exists no generic video analysis technique that successfully deals with a variety of videos that vary depending on the intent of video producer. There are a number of existing video analysis methods with their performance limited on the specific application. Shot detection is used to enhance the performance of video analysis by decomposing a complex video into a number of simple video segments.

2.1. SBD

Video consists of a number of scenes of different content and scene is composed of a number of shots, in which shot is a basic unit of content and consists of a number of frames. Figure 1 shows a hierarchical structure of video: video, scene, shot, and frame. SBD represents finding boundaries of shots, i.e., basic units of content. SBD methods are classified into two categories depending on the features used: frame difference and motion information, each of which will be described in more detail.

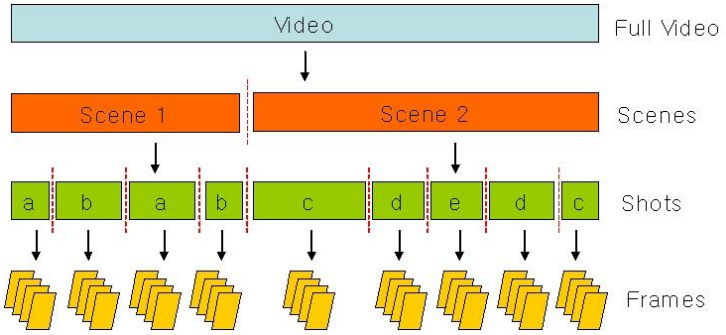


Figure 1. Hierarchical structure of video.

2.1.1. SBD Using Frame Difference Measure

For SBD, the simplest low-level one among various features extracted from input video is the sum of absolute frame differences (FDs) of pixel intensities. Every shot has different objects, people, and background, where shot boundary is defined as the frame at which change occurs. Using this characteristic, the sum of absolute FDs is used to find the shot boundary. Between successive frames k and $k+1$, framewise FD measure $d(k)$ at frame k is defined as [1, 2]

$$d(k)=\sum_{x=0}^{X-1}\sum_{y=0}^{Y-1}\left|I(x,y,k+1)-I(x,y,k)\right| \quad (1)$$

where $I(x,y,k)$ represents intensity at pixel (x,y) at frame k with $X\times Y$ denoting frame size. This measure is computationally fast and gives satisfactory SBD results for video with slow object/camera motion, however, produces incorrect results for video containing fast motion or abrupt illumination change.

To solve the problem, a SBD method using the displaced frame difference (DFD) measure is used. It gives better result robust to motion of objects or people than the method using the FD measure, however, still gives inaccurate result for video with illumination change and abrupt motion.

2.1.2. SBD Using Motion Information

Motion information is another low-level feature for SBD. An improved method [3] uses motion estimation (ME) to avoid performance degradation of SBD methods by motions. This approach considers both object and camera motions, however, still shows degraded SBD performance for

fast-motion video such as sports clip. Other motion-based SBD methods [5], [17], [18] consider dominant or global camera motions and detect shot boundaries by finding abrupt changes in dominant motions. Each shot has relatively similar motions of objects, people, and camera. Usually camera motion is relatively the same between frames within a shot and abruptly changes between shots. This characteristic of content change between shots can be used for SBD, e.g., by block matching algorithm (BMA) [4] and spatial temporal slice [5].

BMA and optical flow can compute motion of small objects or people between video frames, thus they can predict object motion in successive frames and detect shot boundaries as the frames at which abrupt motion occurs. On the contrary, spatial temporal slice method focuses on detecting camera motion. Figure 2 shows SBD computed by the spatial temporal slice method, where the horizontal direction represents time index and each vertical line denotes the intensity of the pixels on the slice lying across the image frames.

Figure 2. SBD using the spatial temporal slice method.

Although motion based SBD methods consider global (camera) or local (object) motions, the performance is sensitive to object or camera motion. The image frames in a video sequence are easily blurred by fast motion, and the conventional features computed from the blurred images are not effective or reliable for determining whether the frame is a shot boundary or not. In video with fast motion of object or people and rapid illumination change, shot boundary is detected incorrectly with large discontinuity measure (computed using low-level image features such as color, edge, shape, motion, or their combinations) similar to that of correct shot boundary. In this case, it is difficult to select an optimum threshold for SBD. For example, if threshold is set to a low value, shot boundaries are incorrectly detected with ambiguous discontinuity measure values (false detection). On the contrary, if threshold is set to a high value, true shot boundaries are missed (miss detection). Also an adaptive SBD algorithm was proposed to detect not only abrupt transitions, but also gradual transitions [16].

In this paper, instead of changing threshold adaptively, our previous SBD approach [12] is used, in which motion based features were developed for SBD especially in the presence of fast moving objects/background. The paper addresses the problem of false/miss detection of shot boundaries in fast-motion video and presents a new algorithm that can overcome the problem. The algorithm reduces the possibility of miss/false detection using the motion-based features: the modified DFD and the blockwise motion similarity.

2.2. Key Frame Detection

Key frame denotes the frame that represents content contained in the shot. Shot is assumed to contain no abruptly appearing or disappearing objects or people. Thus, rather than using all the frames, only a small number of key frames containing distinctive objects or people can be efficiently used for video indexing, which reduces data redundancy and does not require a high computational load.

A simple and common method for selecting key frames [3] is to choose the frame in the middle of the shot. Another method is to find start and end frames of the shot, or to select frames by uniform sampling of frames. However, this method uses fixed sampling interval of frames, thus has a drawback that it does not consider which frame is capable of representing the shot effectively. If the sampling interval is small, many key frames are selected and thus can describe content detail of the video with a high computational load. On the contrary, with a large sampling interval, a small number of key frames may miss the frame containing distinctive objects or people. Various key frame extraction methods for video summary were reviewed and compared based on the method, data set, and the results [6]. A feature-based key frame extraction method for structuring low quality video sequences was presented [7]. Also, a spatial-temporal approach based key frame extraction method was developed [8].

2.3. Video Indexing

Video indexing is defined as the automatic process of content-based classification of video data. In general video indexing [3], video is decomposed into a number of shots by SBD and video indexing is performed by computing the similarity between shots using the similarity of key frames that are representative frames of shots. In computing the similarity between key frames, features such as pixel intensity/color, motion information, shot length, and distance between shot boundaries are used. A video indexing method based on a spatio-temporal segmentation was proposed, in which salient regions are used as scene descriptors for video indexing [9]. A novel active learning approach based on the optimum experimental design criteria in statistics was also presented [10]. The computed similarities between key frames are represented as similarities between shots by a semantic representation method [11].

3. VIDEO INDEXING

This section proposes a video indexing method. For video indexing application, a SBD method for fast-motion video, which was proposed by the same authors [12, 15], is reviewed and its application to video indexing is presented.

3.1. SBD for Fast-Motion Video

This paper uses a SBD method proposed for fast-motion video [15], which employs two features [12]: motion similarity measure and modified DFD that compensates for the FD error caused by object or camera motion. The discontinuity measure used for SBD is a combination of two measures. Two measures are briefly reviewed.

3.1.1. Motion Similarity Measure

Shot boundary is determined as the frame at which motion directions of objects/people and background change abruptly. The same or similar shots contain usually smooth motion flow of objects/people and background whereas different ones have different objects/people and background. In [15], a blockwise motion similarity was proposed by detecting changes in motion directions of the matched macroblocks (MBs) in three successive frames using the BMA.

The similarity measure of the motion direction at MB *i* is defined as

$$MS_i(k) = \begin{cases} 1, & \text{if } \left|\theta_i(k,k+1) - \theta_i(k-1,k)\right| \le Th_\theta \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $\theta_i(k,k-1)$ represents direction angle of MB computed from frames k and $k-1$, and Th_θ denotes the angle threshold. The motion similarity $MS(k)$ at frame *k* is defined as the average of $MS_i(k)$:

$$MS(k) = \frac{1}{N_b}\sum_{i=1}^{N_b} MS_i(k) \tag{3}$$

where $N_b = XY/M^2$ represents the total number of non-overlapping MBs in a frame, with $X \times Y$ and $M \times M$ denoting the frame size and MB size, respectively. The motion similarity $MS(k)$ is larger if more block pairs at frame k have similar motion directions in a shot.

3.1.2. Modified DFD

The DFD has been widely used as a measure of discontinuity, because motion-based difference defined between two successive frames is a more efficient feature than other intensity-based features [13]. The blockwise DFD $d_i(u_i, v_i, k)$ computed by the BMA [4] is defined by

$$d_i(u_i, v_i, k) = \min_{-M \le u, v \le M} \frac{1}{M^2}\sum_{x=0}^{M-1}\sum_{y=0}^{M-1}\left(I_i(x+u, y+v, k+1) - I_i(x, y, k)\right)^2 \tag{4}$$

where $I_i(x,y,k)$ signifies the intensity at (x,y) of $M \times M$ MB *i* at frame *k*, and *u* and *v* denote horizontal and vertical displacements, respectively, with $(2M+1)\times(2M+1)$ search range assumed. The framewise DFD $d(k)$ at frame k is computed by averaging the blockwise DFDs:

$$d(k) = \frac{1}{N_b}\sum_{i=1}^{N_b} d_i(u_i, v_i, k) \tag{5}$$

where u_i and v_i signify the horizontal and vertical components of the MV for MB i, respectively, and N_b denotes the number of MBs at frame k. The DFD value at shot boundary is high whereas low at non-shot boundary. It is a useful feature for video sequences with slow motions of objects/people and relatively uniform illumination. However, the DFD value obtained from fast-motion sequences such as action movies or sports clip is too ambiguous for accurate SBD because

blur produced by fast motion gives large ME errors. Most sequences in fast-motion video are somewhat blurred because the exposure time of video camera is long enough to produce blurring, in which motion blur cannot be ignored.

We assume that the DFD in a motion-blurred image is larger than that in a non-blurred image, and that the larger the motion magnitude, the larger the DFD. To get a DFD value for accurate SBD, a modified DFD [15] is used, in which the motion magnitude term and the distribution of the DFD value that is computed in a sliding window with size of $N+1$ (N: even) is used. First, we simply divide the conventional blockwise DFD $d_i(u_i, v_i, k)$ by the motion magnitude $\sqrt{u_i^2(k)+v_i^2(k)}$ for non-zero MV. The normalized framewise DFD $d^*(k)$ at frame k is defined as

$$d^*(k) = \frac{1}{N_b}\sum_{i=1}^{N_b} d_i^*(k) \quad . \tag{6}$$

In [15], it is observed that the magnitude of the normalized framewise DFD $d^*(k)$ due to fast motion is reduced to some degree, while that due to shot boundaries is not significantly affected. Therefore, it is noted that the normalization decreases ambiguous values that give false classification in fast-motion video and reduces the possibility of false detection at shot boundaries. We can reduce miss detection by choosing shot boundary candidates that satisfy the following condition:

$$d^{**}(k) = \begin{cases} d(k), & \text{if } d(k) = \max\limits_{-N/2 \le j \le N/2} d(k+j) \text{ and } \frac{1}{3}\sum\limits_{j=-1}^{1} d(k+j) \ge \frac{C_1}{N+1}\sum\limits_{j=-N/2}^{N/2} d(k+j) \\ d^*(k), & \text{otherwise} \end{cases} \tag{7}$$

where C_1 is an empirically selected constant. In most of the video sequences, it is found that, at a shot boundary k, the DFD $d(k)$ is usually maximum among the consecutive N frames and the average value of the framewise DFDs in three frames around the shot boundary k is substantially larger than those of N frames. In this case, $d(k)$ is not normalized just to enhance the chance of being detected as a shot boundary.

3.1.3. Discontinuity Measure and Rules for SBD

Through experiments with various test video sequences, both the modified DFD and the motion direction similarity measure are shown to be highly effective for SBD of fast-motion video. Next, a measure of discontinuity based on the combination of the two features [15] is used:

$$Z(k) = \frac{d^{**}(k)}{\frac{1}{N+1}\sum\limits_{j=-N/2}^{N/2} d^{**}(k+j)} \times \exp\left(\frac{-C_2 MS(k)}{\frac{1}{N+1}\sum\limits_{j=-N/2}^{N/2} MS(k+j)} \right) \tag{8}$$

where C_2 is a constant that determines the rate of decrease of $Z(k)$ by the motion similarity at shot boundaries. Equation (8) shows that the discontinuity measure $Z(k)$ is large when two features change abruptly whereas small when they have little change. This different characteristic of the

discontinuity measure can be utilized for separation of shot boundaries from non-shot boundaries and to reduce the ambiguity in SBD. A sliding window is used to consider the temporal variation of the discontinuity measure. We can determine shot boundaries using following detection rules:

$$Z(k) > Z(k+j),\ -N/2 \le j \le N/2,\ j \ne 0 \tag{9}$$

$$Z(k) \ge \frac{C_3}{N+1} \sum_{j=-N/2}^{N/2} Z(k+j) \tag{10}$$

.

If the discontinuity measure at frame *k* satisfies detection rules, it is assumed that an abrupt shot change occurs at frame *k*. The first detection rule in equation (9) finds shot boundary candidates whereas the second one in equation (10) prevents gradual cuts from being selected.

3.2. Key Frame Extraction

Different shots contain different objects, people, and background. Although we cannot recognize them and thus do not know what objects are contained in each shot, we can detect shot boundaries by perceiving the sudden changes of visual content elements. The SIFT [14] matches the query image with one of the training images. In SBD, a training image corresponds to the reference frame and the query image is the current frame. It is assumed that the number of matched points is large between successive frames within the same shot, whereas that at the shot boundary is zero or much smaller than that within the same shot. Thus, the number of matched points is used as a criterion for SBD [15].

The key idea in SBD methods using low-level features is that image attributes or property change abruptly between shots, for example, conventional SBD methods find abrupt changes in color histogram or object/camera motions. In this paper, we investigate the discontinuity of local image features such as those extracted from objects, people, and background at shot boundaries. It is not unreasonable to assume that local image features should be continuously recognizable within the same shot whereas a discontinuity occurs between successive shots. To detect shot boundaries, discontinuities are detected by extracting and matching the SIFT feature points that can efficiently represent objects, people, and background in successive video frames. In this paper, we use the SIFT for establishing feature-based correspondences between two successive frames rather than performing segmentation and matching of objects in successive frames. With many test video sequences, it is observed that the SIFT features are continuously detected from objects, people, and background, and then matched within the same shot, whereas they go through abrupt changes over different shots. We establish feature point correspondences using the point set P_k extracted from the reference frame *k* and the point set P_{k+l} detected at the current frame *k+l*, where *l* denotes the frame distance between the reference and current frames and is to be selected depending on the type of shot boundaries: hard-cut and gradual-transition such as panning, tilting, and fade in/out. Two points are assumed to be correctly matched if a matched point p_{k+l} ($p_{k+l} \in P_{k+l}$) is located within a square window at the current frame *k+l*, where p_k ($p_k \in P_k$) is a center point of a window. For reduction of the computational complexity, the modified SIFT can be used [15].

SBD using SIFT features has a drawback that its performance is degraded for images or videos blurred by fast motion or containing uniform regions such as sky. Also if rotation of objects is larger than some value, e.g., 30 degrees, matching by SIFT features gives incorrect results. This

section presents a key frame extraction method that prevents performance degradation caused by drawbacks of the SIFT.

The amount of motion, obtained in SBD, of objects, people, or background between successive frames can be computed by the magnitude of MV. The framewise motion information $MI(k)$ between successive frames can be defined by

$$MI(k) = \frac{1}{N_b} \sum_{i=1}^{N_b} \sqrt{u_i^{\,2} + v_i^{\,2}} \tag{11}$$

where u_i and v_i signify the horizontal and vertical components of the MV for MB i, respectively, and N_b denotes the number of MBs at frame k.

If motion of objects or people is large, large $MI(k)$ is obtained, otherwise, small $MI(k)$. Figure 3 shows an example of framewise motion information $MI(k)$ in terms of the frame index of test video. The frame that has the minimum $MI(k)$ is selected first and N_k key frames are chosen adjacent to (before or after) the frame that has the minimum $MI(k)$, where N_k is an experimentally selected even number. Thus, $(N_k + 1)$ key frames are defined in each shot. This key frame extraction method can prevent the performance degradation of the SIFT caused by motion blur or rotation of objects. Selection of more than one key frame gives result robust to three-dimensional (3-D) rotation of objects, because if objects or people move, then observation of several successive frames gives 3-D information of objects or people.

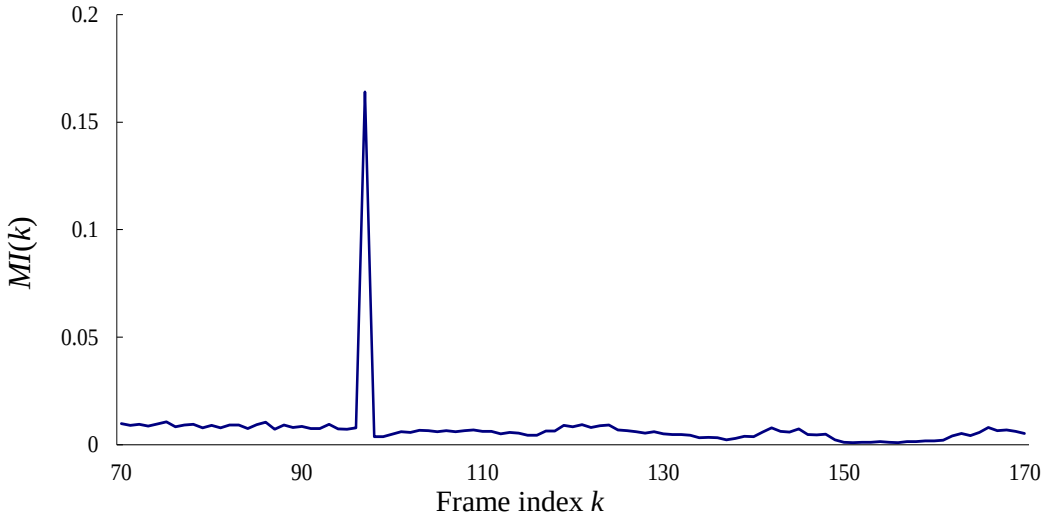


Figure 3. *MI*(*k*) as a function of frame index.

3.3. Video Indexing

This section proposes a video indexing method, in which the SIFT [15] is applied to key frames that are extracted between shot boundaries detected by the SBD method [12]. Video indexing is to automatically classify video content using features extracted from video. As explained previously,

a scene, which is a unit of content representation in video, consists of a number of shots. Figure 4 shows two example structures of scenes consisting of general shots, in which scene 1 consists of shot *a* and shot *b* whereas scene 2 is composed of shot *c*, shot *d*, and shot *e*. Generally, different shots have different objects, people, and background [15], however, within a limited range of frames the same objects or people can be contained in different shots. Also, between different scenes, different objects and people are contained.

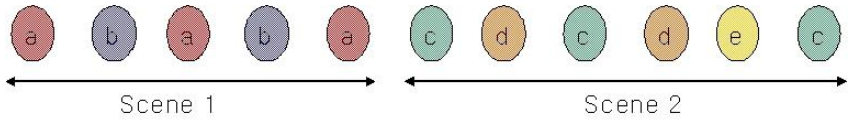


Figure 4. Examples of scene structure.

The SIFT is applied to (N_k+1) key frames to compute the similarity between key frames detected by the SBD method [12], where it is assumed that there exist the same or similar objects, people, or background in two key frames. If the number of matched points (i.e., key points) is larger than threshold, it is assumed that the same or similar objects, people, or background are contained in these key frames. Then, key frames are grouped by the similarity (i.e., by the number of matched points) computed using the SIFT. By grouping similar shots, a scene is constructed.

Using the SIFT, this paper detects different objects and people by comparing SIFT feature points that are contained in key frames within the limited range of shots. The SIFT is robust to scale change of objects, however, is sensitive to motion blur and 3-D rotation of objects. Thus, in each shot, we first detect the key frame with the lowest amount of motion and select N_k adjacent frames, i.e., a total of (N_k+1) key frames are selected in each shot. For fast-motion video, N_k adjacent frames gives blurred or rotated versions of the object in the key frame with the lowest amount of motion, thus similarity computation between shots gives accurate results by using additional N_k frames adjacent to the key frame with the lowest amount of motion. Similarity computation between key frames is performed by matching feature points detected by the SIFT.

Figure 5 shows an example of video indexing. The SIFT is applied to key frame set at shot S (containing N_S key frames) and key frame set at shot $S+l$ (containing N_{S+l} key frames), where $-\frac{N_2}{2} \le l \le \frac{N_2}{2}$ with N_2 being an even constant. The similarity at $(S, S+l)$ is computed using $(N_2 \times N_S \times N_{S+l})$ possible matchings. The relation between shot S and shot $S+l$ is represented by the number of matched key points. If the number is larger than predefined threshold value, shot S and shot $S+l$ are similar shots containing the same or similar objects, people, or background. In this case, two similar shots are grouped into the same scene. Even if two shots are not temporally adjacent (i.e., separated by l), they may contain the same or similar objects, people, or background, as illustrated in Figure 4(b).

Figure 5. Video indexing.

It is difficult to select appropriate value of shot range N_2. For example, if the same objects, people, or background occur frequently in video, large N_2 gives a high probability that a large

number of frames (or a number of the same/similar shots) are merged together into the same scene. Otherwise, a scene contains relatively different shots. In this paper, empirical value of $N_2 = 20$ is used, as used in existing methods.

To explain the procedure of the proposed method more clearly, a pseudo code is shown in the following;

```
Step 1. Shot Change Detection
WHILE with Whole Frames
   Calculate DFD via ME for Shot Change Detection
   Calculate Similarity of Motion Direction with Previous Frame
   IF DFD & Motion Direction Similarity > THRESHOLD_0
      Detect Shot Change Detection
   END IF
   IF MIN(DFD)
      Select Key Frame
   END IF
END of WHILE

Step 2. Descriptor Calculation Using SIFT
WHILE with Whole Key Frames
    Do SIFT
END of WHILE

Step 3. Grouping Shots with Descriptors
WHILE with Whole Key Frames
    WHILE with Key Frames of Each Shot
        Calculate Similarity of Each Key Frame by Comparing Descriptors of Key Frames
        IF Similarity > THRESHOLD_1
           Do Grouping Shots
    END of WHILE
END of WHILE
```

4. EXPERIMENTAL RESULTS AND DISCUSSIONS

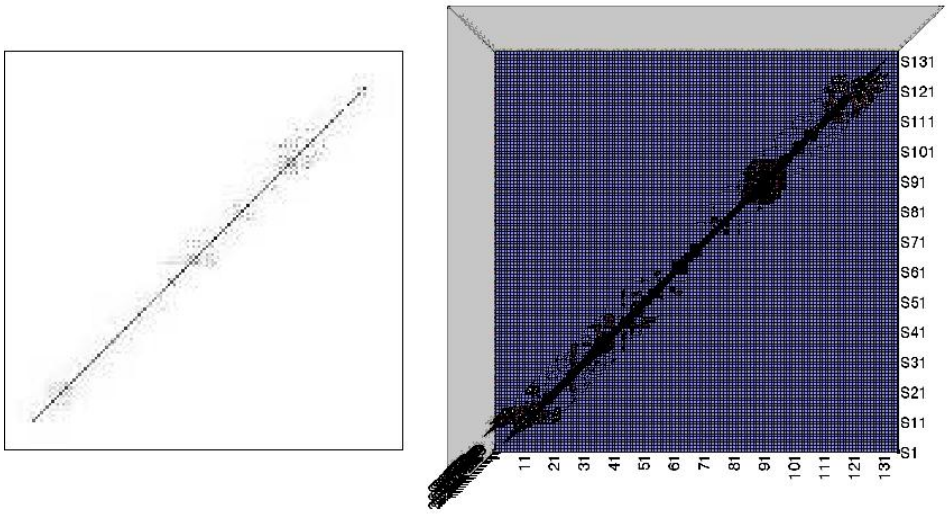
This section presents experimental results and discussions of the proposed video indexing method, in which SBD is performed using our previous SBD method based on blockwise motion-based features [12]. Experimental procedure and results of SBD are described briefly first, followed by experimental results of video indexing.

Our previous SBD method for fast-motion video is applied to seven test videos as used in [12], which consist of 8,000-15,000 frames sampled at 24 frames per second (fps), and a full search BMA is used to estimate blockwise MV. For each sequence, a human observer identifies the shot boundaries as the ground truth. The problem of SBD using low-level image features such as color or motion information is inherently heuristic and it is not easy to mathematically justify the optimal selection of parameter values. With the test video sequences used, the parameter set (C_1, C_2, C_3) that minimizes the detection error is experimentally selected as (3.1, 1.5, 6). The size N_k of a temporal sliding window is defined as the maximum size that does not exceed the average shot length [4], in which the optimal size of N_k =12 is experimentally selected. Note that our previous SBD algorithm gives high detection performances for all test sequences used in

experiments, because of the motion-based discontinuity measure that is suitable for detecting shot boundaries in fast-motion video such as action movies or sports clip [12].

Using SBD results, the performance of the proposed video indexing method is shown. Figures 6-10 show experimental results of five test videos by the proposed video indexing method, in which shot grouping results are illustrated. Horizontal and vertical axes show shot indices, and the number of matched key points between two shots, i.e., the similarity between two shots, is illustrated in two-dimensional (2-D) and 3-D representations. In each figure, similarity between two shots are represented as gray value, e.g., the similarity between shots 10 and 20 is represented as gray level at (10, 20). The darker, the larger the similarity between two shots. Note that dark points gather mostly along the diagonal line, i.e., large similarity exists between adjacent shots.

Video indexing is performed using test videos with shot detection results shown in [12]. For example, test video 1 consists of 131 shots and similarity between each pair of shots is shown in Figures 10(a) and 10(b), in 2-D and 3-D plots, respectively. Large similarity values occur along the diagonal line, i.e., adjacent shots have large similarity values. As observed from the 2-D and 3-D plots of similarity, shots between 85 and 95 have large similarity values. If the similarity value is larger than the pre-specified threshold value, then these 11 shots can be combined together as the same scene as shown in Figure 4. Representation of scene structure of test video 1 depends on the parameter values such as the number of adjacent frames in key frame detection N_k and the threshold for similarity values between a pair of shots.

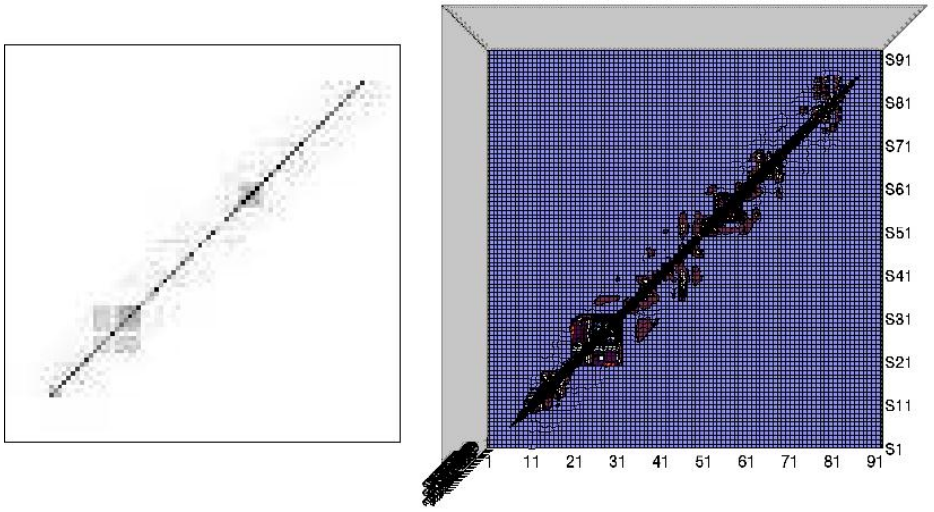


(a) (b)

Figure 6. Result of video indexing (test video 1). (a) similarity between two shots are represented as gray level. (b) 3-D plot of the similarity between two shots.

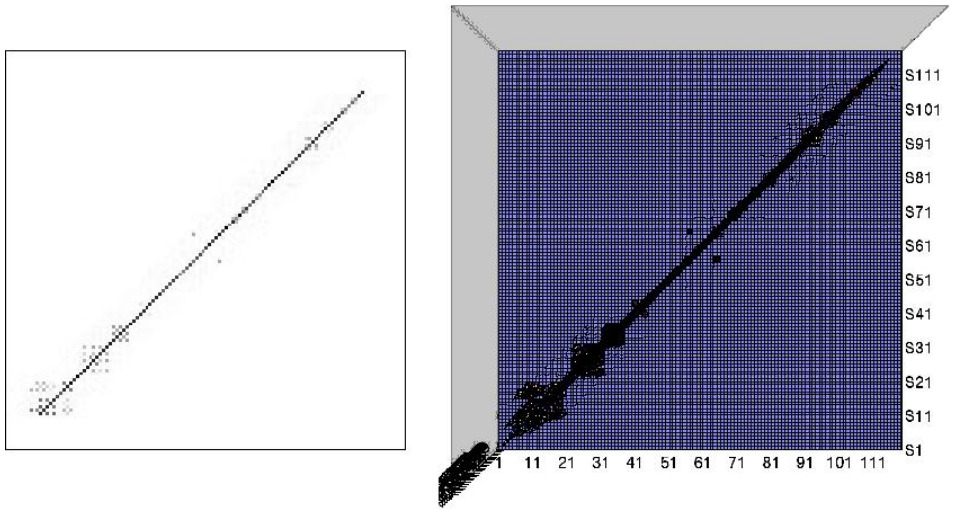
Similarly, video indexing results with test videos 2, 3, 4, and 5 are shown in Figures 7, 8, 9, and 10, respectively, in 2-D and 3-D representations. Test videos 2, 3, 4, and 5 consist of 91, 111, 71, and 131 shots, respectively, in which the number of shots depends on the characteristics of the test video and parameter values used in SBD. Final scene structure of the test video depends on the parameter values used in video indexing. Note that the similarity values along the diagonal

lines have different cluster patterns depending on the characteristics of the test video, i.e., illustrating different scene structure of different test videos.



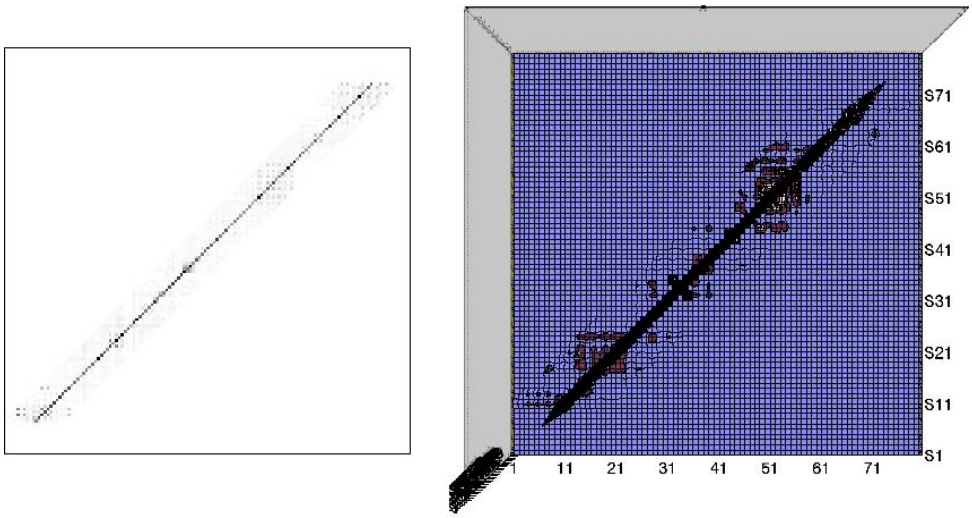
(a) (b)

Figure 7. Result of video indexing (test video 2). (a) similarity between two shots are represented as gray level. (b) 3-D plot of the similarity between two shots.



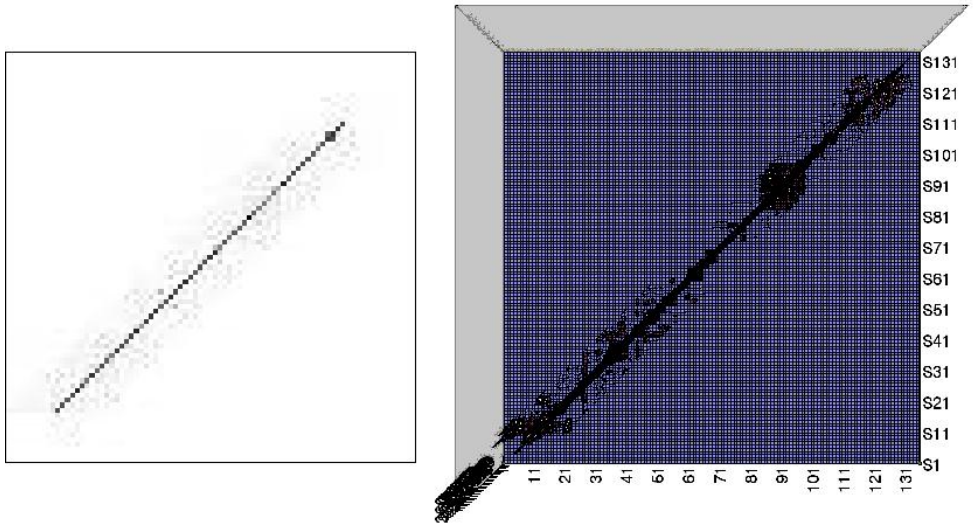
(a) (b)

Figure 8. Result of video indexing (test video 3). (a) similarity between two shots are represented as gray level. (b) 3-D plot of the similarity between two shots.



(a) (b)

Figure 9. Result of video indexing (test video 4). (a) similarity between two shots are represented as gray level. (b) 3-D plot of the similarity between two shots.



(a) (b)

Figure 10. Result of video indexing (test video 5). (a) similarity between two shots are represented as gray level. (b) 3-D plot of the similarity between two shots.

Different video indexing methods have many heuristic parameters, depending on characteristics of test video and applications. Due to different heuristics used in each video indexing method, it is difficult to compare experimental results of the proposed video indexing method with existing video indexing methods.

In the proposed algorithm, computational steps such as ME, detection of SIFT feature descriptors, and matching between SIFT descriptors are performed. Among them the most computationally

expensive step is ME, in which computational complexity is O(n), where n is the number of frames in video.

In summary, this paper proposes a video indexing method suitable for fast-motion video with motion blur or rotation of objects. For SBD, a number of frames adjacent to the key frame with the lowest amount of motion are used in the SIFT feature matching for robustness to motion blur due to fast motion or to 3-D rotation of objects in frames.

5. CONCLUSIONS

This paper proposes an efficient video indexing method for fast-motion video. First, an SBD method robust to fast motion is used, in which two motion-based features such as the modified DFD based on the BMA and the blockwise motion similarity are used. Then, a number of key frames are detected in each shot and applied to video indexing using the SIFT feature matching, which is robust to size and illumination change. The proposed video indexing algorithm gives good results because SBD based on motion similarity information and modified DFD is performed by considering motion of objects, people, and background. Future research will focus on developing a SBD method that is effective for different types of shot boundaries and on the investigation of a shot grouping method that effectively summarizes the fast-motion content.

REFERENCES

[1] Z. Rasheed and M. Shah, "Scene boundary detection in Hollywood movies and TV shows," in Proc. Computer Vision Pattern Recognition, vol. 2, pp. 343-348, Madison, WI, USA, June 2003.

[2] H. Liu, W. Meng, and Z. Liu, "Key frame extraction of online video based on optimized frame difference," in Proc. 2012 9th Int. Conf. Fuzzy Systems and Knowledge Discovery, pp. 1238-1242, 2012.

[3] W. Hu, N. Xie, L. Li, X. Zeng, and S. Maybank, "A survey on visual content-based video indexing and retrieval," IEEE Trans. Systems, Man, and Cybernetics–Part C: Applications and Reviews, vol. 41, no. 6, pp. 797-819, Nov. 2011.

[4] F. Moschetti, M. Kunt, and E. Debes, "A statistical adaptive block-matching motion estimation," IEEE Trans. Circuits Systems for Video Technology, vol. 13, no. 5, pp. 417-431, May 2003.

[5] C. W. Ngo, T. C. Pong, and H. J. Zhang, "Motion analysis and segmentation through spatio-temporal slices processing," IEEE Trans. Image Processing, vol. 12, no. 3, pp. 341-355, Mar. 2003.

[6] C. Sujatha and U. Mudenagudi, "A study on key frame extraction methods for video summary," in Proc. 2011 Int. Conf. Computational Intelligence and Communication Systems, pp. 73-77, Gwalior, India, Oct. 2011.

[7] P. Kelm, S. Schmiedeke, and T. Sikora, "Feature-based video key frame extraction for low quality video sequences," in Proc. 10th Workshop Image Analysis for Multimedia Interactive Services, pp. 25-28, London, United Kingdom, May 2009.

[8] Z. Sun, K. Jia, and H. Chen, "Video key frame extraction based on spatial-temporal color distribution," in Proc. Int. Conf. Intelligent Information Hiding and Multimedia Signal Processing, pp. 196-199, Harbin, China, Aug. 2008.

[9] M. Sameh, S. Wided, A. Beghdadi, and C. B. Amar, "Video indexing using salient region based spatio-temporal segmentation approach," in Proc. 2012 Int. Conf. Multimedia Computing and Systems, pp. 170-173, Tangier, Morocco, May 2012.

[10] Z.-J. Zha, M. Wang, Y.-T. Zheng, Y. Yang, R. Hong, and T.-S. Chua, "Interactive video indexing with statistical active learning," IEEE Trans. Multimedia, vol. 14, no. 1, pp. 17-27, Feb. 2012.

[11] S. Zhu and Y. Liu, "Scene segmentation and semantic representation for high-level retrieval," IEEE Signal Processing Letters, vol. 15, no. 1, pp. 713-716, 2008.

[12] M.-H. Park, R.-H. Park, and S. W. Lee, "Efficient shot boundary detection for action movies using blockwise motion-based features," Lecture Notes in Computer Science, Advances in Visual

Computing: First Int. Symposium, ISVC 2005, Eds. G. Bebis et al., vol. 3804, pp. 478-485, Dec. 2005.

[13] J.-Y. Kim, R.-H. Park, and S. Yang, "Block-based motion estimation using the pixelwise classification of the motion compensation error," in Proc. 2005 Digest of Technical Papers Int. Conf. Consumer Electronics, pp. 259-260, Las Vegas, NV, USA, Jan. 2005.

[14] D. G. Lowe, "Distinctive image feature from scale-invariant keypoints," Int. Journal of Computer Vision, vol. 60, no. 2, pp. 91-110, Nov. 2004.

[15] M.-H. Park, R.-H. Park, and S. W. Lee, "Shot boundary detection using scale invariant feature transform," in Proc. SPIE Visual Communications and Image Processing 2006, pp. 478-485, San Jose, CA, USA, Jan. 2006.

[16] Y. Huo, P. Zhang, and Y. Wang, "Adaptive threshold video shot boundary detection algorithm based on progressive bisection strategy," Journal of Information & Computational Science, vol. 11, no. 2, pp. 391-403, Jan. 2014.

[17] D. J. Lan, Y. F. Ma, and H. J. Zhang, "A novel motion-based representation for video mining," in Proc. Int. Conf. Multimedia and Expo, vol. 3, pp. 469-472, Baltimore, MD, USA, 2003.

[18] Y. F. Ma and H. J. Zhang, "Motion texture: A new motion based video representation," in Proc. Int. Conf. Pattern Recognition, vol. 2, pp. 548-551, Quebec, QU, Canada, 2002.

Authors

Min-Ho Park received the B.S. degree in electronic engineering from Dongguk University, Seoul, Korea, in 2004. He received the M.S. degree in electronic engineering from Sogang University, Seoul, Korea, in 2006. In 2006, he joined video team in the Research & Development of PIXTREE, Inc. as senior engineer. He has been a senior software engineer in SmartTV Platform team in the Research & Development of LG Electronics Inc. since 2010. His research interests include video processing, analysis, indexing, and compression.

Rae-Hong Park received the B.S. and M.S. degrees in electronics engineering from Seoul National University, Seoul, Korea, in 1976 and 1979, respectively, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1981 and 1984, respectively. In 1984, he joined the faculty of the Department of Electronic Engineering, Sogang University, Seoul, Korea, where he is currently a Professor. In 1990, he spent his sabbatical year as a Visiting Associate Professor with the Computer Vision Laboratory, Center for Automation Research, University of Maryland at College Park. In 2001 and 2004, he spent sabbatical semesters at Digital Media Research and Development Center (DTV image/video enhancement), Samsung Electronics Co., Ltd. In 2012, he spent a sabbatical year in Digital Imaging Business (R&D Team) and Visual Display Business (R&D Office), Samsung Electronics Co., Ltd. His current research interests are video communication, computer vision, and pattern recognition. He served as Editor for the Korea Institute of Telematics and Electronics (KITE) Journal of Electronics Engineering from 1995 to 1996. Dr. Park was the recipient of a 1990 Post-Doctoral Fellowship presented by the Korea Science and Engineering Foundation (KOSEF), the 1987 Academic Award presented by the KITE, the 2000 Haedong Paper Award presented by the Institute of Electronics Engineers of Korea (IEEK), the 1997 First Sogang Academic Award, and the 1999 Professor Achievement Excellence Award presented by Sogang University. He is a co-recipient of the Best Student Paper Award of the IEEE Int. Symp. Multimedia (ISM 2006) and IEEE Int. Symp. Consumer Electronics (ISCE 2011).