Paper

Two-Step Detection of Video Shot Boundaries in a Wavelet Transform Domain

Satoshi HASEBE*, Makoto NAGUMO**,
Shogo MURAMATSU*, Hisakazu KIKUCHI (Member)*
*Department of Electrical and Electronic Engineering, Niigata University
**Broadcasting System of Niigata, Inc.

⟨Summary⟩ This paper presents a two-step algorithm for detecting video shot boundaries in a wavelet transform domain. It captures gradual shot transitions such as dissolves and fades as well as abrupt shot transitions such as cuts. At the first step, it computes a distance between video intervals to find an isolated interval. Then, at the second step, it computes a distance between frames to specify the exact location of a shot boundary. Its effectiveness is evaluated in terms of recall and precision.

Key words: Video analysis, video indexing, shot boundary detection, wavelets.

1. Introduction

Videos have become major contents in the areas of telecommunications, entertainment and education. A variety of moving pictures are available on the Internet, such as video clips, movie pictures and live TV programs. Recent developments in video coding techniques and network technology have made it possible to access a huge number of videos and streaming videos over networks. More and more video contents are available on line. On the other hand, non-linear editing of videos has come into wide use.

To deal with vast amount of visual information, effective indexing techniques are needed[1),2)]. MPEG-7 is initiated for the purpose of providing multimedia description interfaces and useful tools for describing video contents[3)].

The goal of this study is to develop a framework of automated video indexing for browsing a video quickly, retrieving similar videos by their contents, and aiding a user to edit a video. We have proposed a video querying technique that uses an image or a sequence of frames as a query to retrieve similar parts of a video[4)]. It is important for content-based video indexing to decompose a video sequence into appropriate temporal segments. A shot is considered as the most basic structure of a video, and is qualified for the purpose of video indexing.

In this paper, a two-step algorithm for detecting video shot boundaries is presented. It captures gradual shot transitions such as fades and dissolves as well as abrupt transitions such as cuts. It computes a distance between video intervals and that between frames by comparing feature vectors. A feature vector comprises the coarsest subband and the significance map of the finer subband of the wavelet transform of a frame or an interval. The effectiveness of the proposed method is evaluated in terms of recall and precision values.

The rest of this paper is organized as follows. Section 2 describes shot transitions and early studies to detect them. Section 3 details the proposed wavelet domain shot boundary detection. Section 4 shows implementations and experimental results. Finally, section 5 states the concluding remarks of this paper.

2. Shot Boundary Detection

2.1 Shot Transitions

A shot is an interval that represents continuous action captured by a single camera[5)]. Two types of shot transitions are considered: abrupt transitions and gradual transitions.

A cut is an abrupt transition that occurs when one shot follows another without any editing effect.

Fading effects are typical gradual transitions. A fade-out is an effect that the luminance of a frame gradually changes leading to a black, white or other homogenous color frame. A fade-in is the reverse process: starting from a homogenous color frame, the frame varies until a definite frame of the following shot appears.

A dissolve is considered as a cross-fading effect. One shot fades out while at the same time the following shot fades in. The dissolve smoothly concatenates two different shots, and it is frequently used in various videos.

A wipe is an impressive gradual transition. Unlike a dissolve, a wipe does not produce a mixture frame of different shots. Instead, one frame of a shot is moving away, and simultaneously another frame in the next shot is sliding in. In another words, a part of a frame is occupied by one shot and the rest of the frame is occupied by the next shot.

2.2 Frame-Based Detection Techniques

A typical shot boundary detection technique computes a distance between successive frames to find a shot transition where some distance becomes significantly large.

A number of automatic shot boundary detection techniques have been proposed[6),7)]. Boreczky[8)] compared several detection algorithms. They are based on pixel differences, histograms, motion vectors and DCT coefficients. Gargi[9)] evaluated several methods that use color histograms, block motion matching and MPEG compressed data.

A pixel difference is the earliest approach to detect a cut in digital video sequence. It is easy to implement and fast. However, it is not robust to some changes other than a cut such as lighting changes, camera motions and noises.

To give a robustness to such changes, a histogram-based approach is developed. Many variations of histogram-based detection techniques are proposed. Global histograms, local histograms and running histograms are investigated by Boreczky[8)], and they are relatively accurate and consistent in the literature.

2.3 Gradual Transition Detection

Frame-based methods work well for detecting abrupt transitions since the involved statistics change dramatically at a cut point. It is, however, sometimes insufficient for detecting a gradual transition. The frame content over a gradual transition is considered as a mixture of entirely different two contents. Thus, the distance is relatively large compared with that in a stable shot. It is, at the same time, not so large compared with that in an abrupt transition since the frame content correlates to some extent with the beginning part and the ending part of the gradual transition. As a result, the frame-based detection approach is likely to miss those shot boundaries, or it can detect so many false shot boundaries.

Zhang[10)] has introduced a twin-comparison approach for detecting gradual transitions. It uses two threshold values: a higher threshold value for detecting a cut, and a lower threshold value for detecting a potential start point of a gradual transition. It first computes a frame distance based on a color histogram comparison. If the distance exceeds the higher threshold value, a cut is declared. If the distance does not exceeds the higher threshold value, but exceeds the lower threshold value, a potential start point of gradual transition is marked, and accumulated distance is calculated until a frame distance drops below the lower threshold value. If an accumulated distance exceeds the higher threshold value, an end point of gradual transition is declared.

Xiong[11)] has developed a *step-variable* method. It reduces computational costs by varying a duration of two frames of which frame distance is calculated. They use two types of distance metrics: a net comparison[12)], that compares the pixels along predefined net lines of an image, and an edge based distance. The net comparison is a pixel difference metric, thus, it is sensitive to luminance changes. To the contrary, the edge based distance is considered to insensitive to luminance changes.

2.4 Compressed Domain Approaches

Since most of videos are popular in the compressed format, it is quite interesting to detect shot boundaries in a compressed domain.

Nang[13)] has proposed a simple shot boundary detection technique that detects a cut from a MPEG video sequence. It compares a type of macroblocks in B-frames to find a significant change. Jun[14] has developed a dissolve detection algorithm that also uses macroblock information. It analyzes the ratio of forward macroblocks in B-frames and the spatial distribution of macroblocks.

Bescos[15] has studied eight metrics for both pixel-domain and compressed-domain, where compressed-domain means DC image of a frame in a MPEG-2 video sequence. A problem of shot boundary detection is modeled as a classification in a decision space. The decision space is a vector space yielded by the application of a single or multiple metrics. As a result of analysis of the maximum capability of separation, an optimal decision space is generated for each type of transitions such as cuts and gradual transitions.

Motion-JPEG 2000[16] is a one of up-coming image and video coding standards. It has a lot of novel features such as quality scalabilities, ROIs (Region Of Interests), and seamless lossy and lossless codings. The fact that Motion-JPEG 2000 involves merely intra-frame coding techniques matches various kinds of applications, e. g. high quality video editing and mobile communications where the mobile terminals have limited CPU resources for motion-compensated video codings. JPEG 2000[17] is a wavelet-based coding technique. Two-dimensional wavelet coefficients are obtained from a partially decoded bitstream.

The proposed two-step shot boundary detection algorithm works on a wavelet transform domain. It generates two different types of metrics from the coarsest and finer subbands, respectively. A combination of these metrics leads to robust detection of both abrupt and gradual shot transitions. In addition, compressed-domain processing will contribute to reduce computations as well as to save a memory.

3. Wavelet Domain Shot Boundary Detection

3.1 Video Interval in a Wavelet Domain

When an image is transformed into a two-dimensional wavelet transform domain, the energy of the image concentrates into lower frequency bands. The coarsest subband component in the two-dimensional spatial wavelet transform looks like a miniature copy of the original image. Also, it is a fact that successive frames over a short period strongly correlate to each other. When a video sequence is transformed into a three-dimensional wavelet transform domain, the video contents are decomposed into subbands with respect to scales in time as well as spatial scales. The coarsest subband component with respect to time scale represents an average over time, and is considered to represent the average information over a particular period. On the other hand, finer subband components represent local and temporal activities over the period. It is quite reasonable to expect that the energy of a video sequence will concentrate into lower frequency bands.

Fig. 1 illustrates a sample of a video interval. It is a part of a video sequence, and consists of several

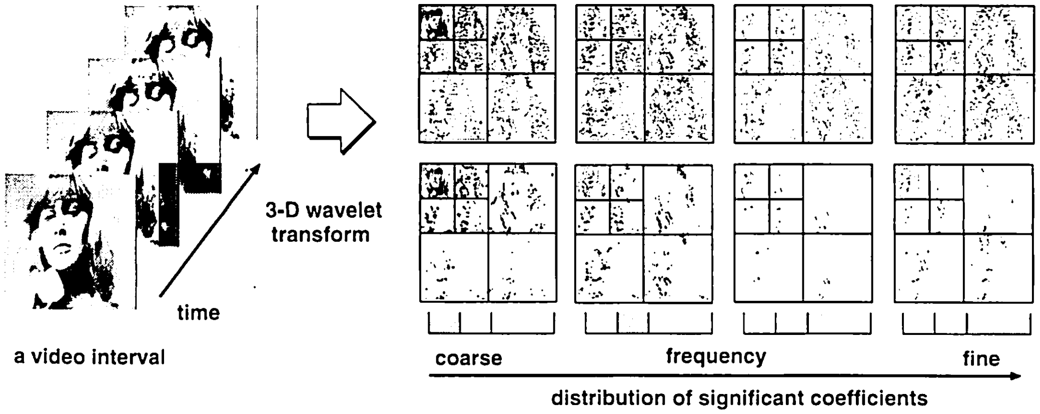

Fig. 1 Example of a video interval (left), corresponding three-dimensional wavelet transforms (top right) and the distribution of significant wavelet coefficients (bottom right)

successive frames, as shown at the left. Its three-dimensional wavelet transform is shown in the top row at the right. The distribution of significant coefficients is in the bottom row at the right, where *significant* implies to be large in magnitude. It should be noted that the leftmost two pictures of the wavelet transform and the significance distribution show the coarsest subband components with respect to the temporal frequency rather than the spatial frequency. The spatially coarsest subband is placed at the top-left corner of individual pictures as usual as in the conventional two-dimensional wavelet transforms of images. One can see, in the figure, that significant coefficients mainly populate in coarser subbands in time as well as in space.

Owing to the above observations, some particular subbands, especially coarser components, in the three-dimensional wavelet transform of a video sequence are assumed to convey significant contents in a video, while significant coefficients in finer subbands imply the presence of edges, textures and sharp variations in time.

3.2 A Metric for Frame Distance

3.2.1 Feature Vector of a Frame

In order to measure the similarity between successive two frames, we have constructed a feature vector for a video frame as follows. Fig. 2 shows the procedure for making a feature vector.

The coarsest subband coefficients have a large average value and a large variance. They are all significant coefficients, and higher precision should be given for their representation. On the other hand, coefficients in the finer subbands have almost zero-mean and a smaller variance. It is more important to describe where a significant coefficient is located in particular finer subbands rather than how large the significant coefficient itself is. Thus, we represent the finer subbands as a binary map. Significant and insignificant coefficients are encoded into one and zero, respectively.

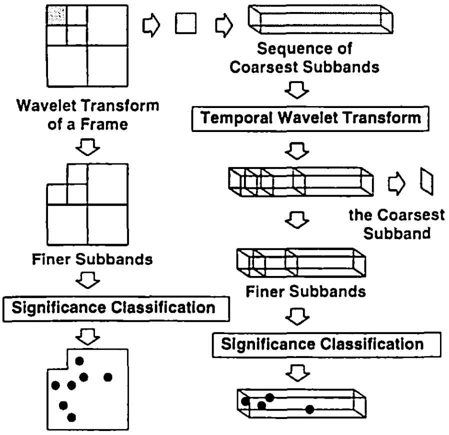
Sequence of Coarsest Subbands
Wavelet Transform of a Frame
Temporal Wavelet Transform
the Coarsest Subband
Finer Subbands
Finer Subbands
Significance Classification
Significance Classification

Fig. 2 Procedure for making significance maps

We are going to classify wavelet coefficients in finer subbands by their significance. One way to do this is thresholding. However, a difficulty arises in defining an appropriate threshold value. Few coefficients can be classified as significant under an excessively high threshold value. On the other hand, most of the coefficients can be classified as significant under an excessively low threshold value. Thus, we take a top-N approach[18),19)]. For a given constant N, largest N coefficients in magnitude are classified as significant and the other insignificant coefficients are neglected. Every coefficient in the finer subbands is quantized into one or zero depending on its significance or insignificance to form a binary representation of a significance map. As a result, a feature vector that consists of the significance map and the coarsest subband coefficients are obtained for a frame in a video sequence.

3.2.2 Definition of a Frame Distance

A feature vector F comprises the coarsest subband C and the significance map S of finer subbands of the two-dimensional wavelet transform of a frame picture.

$$F=\{C, S\}. \qquad (1)$$

The coarsest subband consists of quantized wavelet coefficients. It is a coarse approximation of a frame. The significance map is a binary map: significant coefficients in finer subbands are encoded as unity and insignificant coefficients are encoded as zero. It implies the presence of sharp changes such as edges and textures.

The distance between feature vectors of successive two frames, say F_{n+1} and F_n, is defined after some preliminary definitions.

We measure the similarity between two coarsest subbands, C_{n+1} and C_n, by the L 1 distance, as follows.

$$\|C_{n+1}-C_n\|_{L1}=\sum_i\sum_j|c_{n+1}(i,j)-c_n(i,j)|, \tag{2}$$

where $c(i,j)$ denotes a coefficient at (i,j) in the coarsest subband. The shorter L 1 distance means the closer similarity.

We measure the similarity between two significance maps, S_{n+1} and S_n, by the Hamming distance. The Hamming distance between two binary sequences that have the same length is computed by counting the binaries which differ to each other. It is defined by

$$\|S_{n+1}-S_n\|_H=\sum_i\sum_j\{s_m(i,j)\oplus s_n(i,j)\} \tag{3}$$

where $s(i,j)$ denotes a binary at (i,j) and $\oplus$ represents exclusive OR. The shorter Hamming distance means a better match.

The above two sorts of similarity are combined to define the similarity distance between two feature vectors of frames, F_{n+1} and F_n. It is defined by a weighted sum of the Hamming distance and the L 1 distance, as follows.

$$\|F_{n+1}-F_n\|=w_0\|C_{n+1}-C_n\|_{L1}+w_1\|S_{n+1}-S_n\|_H, \tag{4}$$

where w_0 and w_1 are weights.

3.3 Two-Step Approach for Gradual Transition Detection

3.3.1 Definition of an Interval Distance

In order to capture a gradual shot transition, we divide a given video sequence into multiple intervals and measure the similarity between those intervals. It is desirable that the duration of an interval is long enough to cover a gradual shot transition, and is also short enough so that two or more shot transitions do not appear in a single video interval. These video intervals can have different durations in a single video sequence. Unfortunately, there is no way to determine the optimal duration unless extra knowledge of the video sequence is given. A predetermined regular duration would be a reasonable choice.

The duration of a video interval is arbitrary. The computational cost can, however, restrict the upper bound. Meng[20] detects a dissolve region based on an observation that a dissolve usually lasts from 30 to 60 frames, i.e., about one or two second. Also, Jun[14] applies a heuristic rule that the duration of a dissolve transition is typically longer than 0.3 second.

To measure the similarity between two intervals, an interval distance is defined as follows. In the process of making the feature vector of a frame, the coarsest subband information is stored. A temporal sequence of the coarsest subbands is further decomposed by a one-dimensional temporal wavelet. Every coefficient in finer subbands is encoded into 1 bit to form a significance map. As a result, a feature vector of a given video interval is produced.

The distance between two intervals is computed by comparing feature vectors of those intervals. It is worth noting that Eq.(4) equally applies to the similarity distance between two intervals except that three-dimensional indices are used instead of two-dimensional indices of C and S in Eq.(2) and Eq.(3), respectively.

Fig. 3 plots the frame distance and interval distance for a test video sequence, *Color Harmony for Your Home*. Let f_m be the mth frame. The mth frame distance denotes the similarity distance between f_m and f_{m+1}. Similarly, let I_n be the nth interval. The nth interval distance denotes the similarity distance between I_n and I_{n+1}. Every interval has the same duration of 1.0 second, and the frame rate of the test sequence is 30 frames per second. Thus, I_n contains 30 frames ranging from 30×nth frame through 30×(n+1) th frame.

A cut appears at the 1,673 th frame. The frame distance well captures the cut. A dissolve appears from 1,780 th frame through 1,814 th frame. Unlike a cut, the frame distance does not clearly capture the event of a dissolve. Although, relatively large frame distances are observed around the dissolve, they are

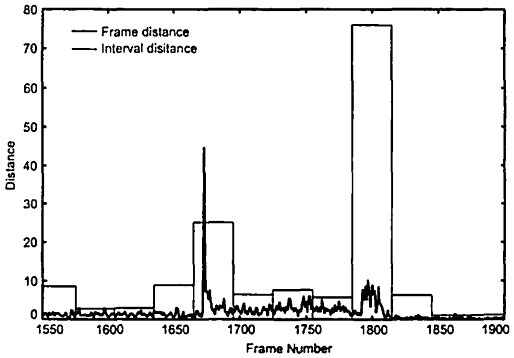

Fig. 3 Frame distance and interval distance for *Color Harmony for Your Home*

not so definite like an impulsive frame distance at a cut point. On the other hand, the corresponding interval distance clearly reflects the presence of the dissolve.

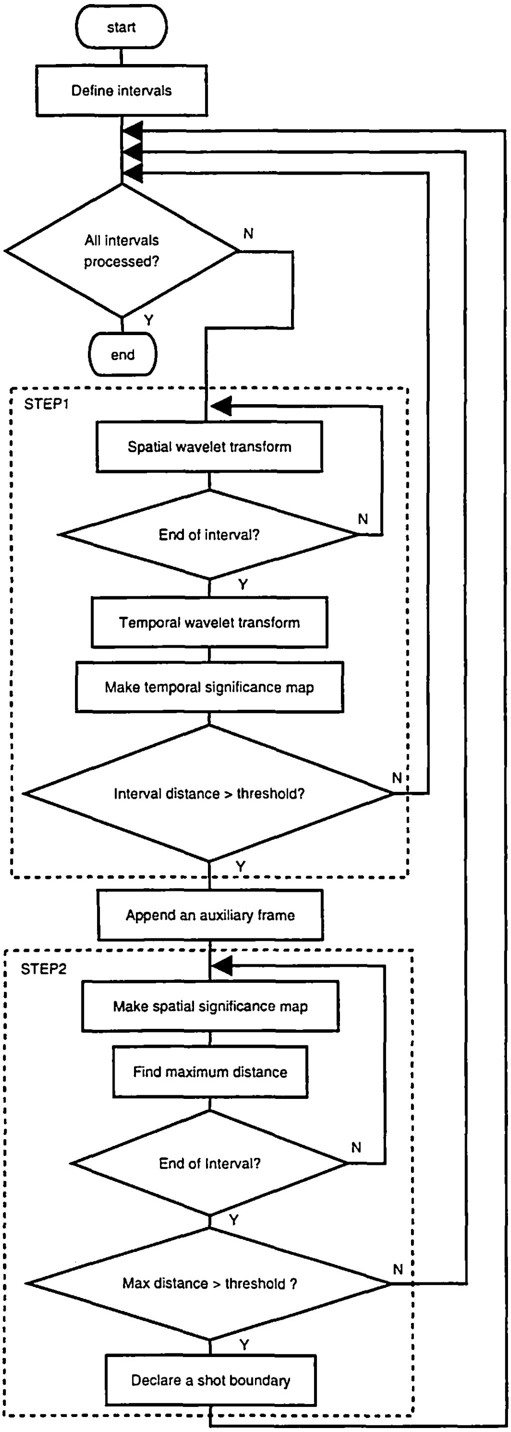
start
Define intervals
All intervals processed?
N
Y
end
STEP1
Spatial wavelet transform
End of interval?
N
Y
Temporal wavelet transform
Make temporal significance map
Interval distance > threshold?
N
Y
Append an auxiliary frame
STEP2
Make spatial significance map
Find maximum distance
End of Interval?
N
Y
Max distance > threshold ?
N
Y
Declare a shot boundary

Fig. 4 Two-Step Algorithm

3.3.2 Two-Step Detection Algorithm

Fig. 4 describes the two-step detection algorithm. A given video sequence is divided into multiple intervals. Let I_n represent the nth interval. I_{n+1} represents the interval that follows the previous interval I_n.

In STEP 1, the interval distance $\|I_{n+1}-I_n\|$ for I_n in issue is computed. If $\|I_{n+1}-I_n\|$ exceeds a given threshold value, an extended interval I'_n is defined by concatenating the first frame in I_{n+1} to the original frames in I_n. Then, STEP 2 is performed for I'_n. The reason for using I'_n instead of I_n is to detect a shot boundary that coincides with the boundary between I_n and I_{n+1}. If $\|I_{n+1}-I_n\|$ does not exceed the threshold value, we conclude that there is no shot transition in I_n, and STEP 1 is performed for the next pair of intervals.

In STEP 2, for every frame in I'_n, a frame distance is computed. Then, a position that takes the maximum distance is searched. If the maximum distance exceeds a prescribed threshold value, a shot boundary is assumed to have been detected. Otherwise, no shot boundary is detected in I'_n and return to STEP 1.

A list of shot boundaries is obtained after the above iterative procedure. The frame distance and the interval distance are computed by comparing the feature vectors for frames and for intervals, respectively. It should be noted that a feature vector is not generated for all frames in a given video sequence. Actually, a frame in an interval that is considered to be unchanged is never processed in STEP 2.

4. Implementations and Evaluations

4.1 Implementations of Detection Algorithms

At first, we have implemented the proposed detection algorithm in the following configurations. For every frame in a given video interval, seven-level spatial decompositions for both rows and columns are performed. The finest two levels are truncated. A sequence of the spatially coarsest subbands is maximally decomposed along time until the temporally coarsest subband component becomes single.

Every frame in a given video interval is transformed into a two-dimensional wavelet transform domain as follows. A one-dimensional wavelet transform is implemented by a lifting scheme[21]. It is known that any wavelet transform implemented by two-band filtering with FIR filters is decomposed into lifting steps[22]. The lifting allows us to implement a wavelet transform in the in-place computation. No additional memory is needed for the computation of a wavelet transform. In addition, filters used in the lifting steps are simple compared with those of filter-banks. Thus, it reduces the computational complexity; asymptotically it is twice as fast as the convolution implementations[22]. A two-dimensional spatial wavelet transform is obtained by alternating horizontal one-dimensional wavelet transforms and vertical transforms.

A choice of a particular wavelet would affect the effectiveness and computational complexity in detecting video shot boundaries. We chose the 5/3-tap integer wavelet as the spatial wavelet, which is mandatory in JPEG 2000. This wavelet would be preferable to describe major contents of a frame image. Generally, frames in a video sequence that exist close in time are likely to be very similar to each other. Especially in a single shot, the background would be almost the same. The Haar wavelet can well describe piece-wise constant signals[23], and is thus appropriate for the temporal wavelet. Moreover, the Haar wavelet is very simple, easy to implement, and has the least computational complexity. Both spatial and temporal wavelets are implemented by the lifting scheme.

Coefficients in finer subbands are sorted by magnitude in decreasing order. Top 50 coefficients are encoded as significant. A list of their indices forms a significance map. A distance between frames and a distance between video intervals are computed by Eq. (4). w_0 is set to be the reciprocal of the number of coefficients in the coarsest subband and w_1 is set to be unity in our experiments. The duration of the video interval used for two-step shot boundary detection is 1.0 second. Those have been computed for each color component of a given video sequence. Finally, a threshold value T_I for an interval distance is given by

$$T_I = wT_F \quad (5)$$

where, T_F denotes a given threshold value for a frame distance and w is a weight. This weight is fixed at 1.25 through an analysis of a few-minute subset of the test video sequence.

Two other detection algorithms are selected for a comparison purpose. A *color code histogram* algorithm has been constructed as follows. After decoding a given video sequence, RGB color frame image is obtained. Each color band is assumed to have 8-bit color depth. For every color band, the most significant 2 bits are taken and they are combined to form a 6-bit color code. A 64-bin color code histogram is calculated for a given frame. A distance between two color code histograms is defined as the sum of absolute difference between corresponding bins.

A twin-comparison approach, the third algorithm, is a device for detecting gradual shot transitions. It is developed by Zhang[10] and is described in Section 2.3. We have implemented it based on above mentioned color code histogram. A lower threshold value T_L is given by

$$T_L = wT_H \quad (6)$$

where, T_H denotes a given higher threshold value and w is a weight. Three weights of 0.5, 0.25 and 0.1 are tested and the best result is used for comparisons.

All test video sequences are encoded with MPEG-1 format. They are transcorded into Motion-JPEG and Motion-JPEG 2000, respectively. Every algorithm is implemented by Java and runs on a personal computer equipped with Intel Pentium 4 CPU. The proposed algorithm runs faster than real time for Motion-JPEG sequences. However, it runs slower than real time for Motion-JPEG 2000 because entropy decoding of Motion-JPEG 2000 requires intensive computations.

4.2 Recall and Precision

We have calculated recall and precision[24] for several test video sequences for evaluations of the proposed method. They are selected from the Open Video Project[25] and MPEG-7 Contents Set[26]. They include various genres such as news, sports, and animations. Ground truth data of shot boundaries are available for some of the test sequences. For the other test sequences, we have manually checked them and constructed the ground truth data.

Recall expresses the percentage of actually detected correct shot boundaries out of the full set of correct shot boundaries. It is defined as

$$r=\frac{n_c}{n_c+n_m}, \tag{7}$$

where n_c is the number of shot boundaries that are correctly detected and n_m is the number of shot boundaries that are missed. Precision expresses the percentage of detected correct shot boundaries among all detected shot boundaries. It is defined as

$$p=\frac{n_c}{n_c+n_f}, \tag{8}$$

where n_f is the number of shot boundaries that are falsely detected.

A recall-precision curve is parameterized by a threshold value used in the detection algorithms. A higher threshold value would produce a higher precision with a lower recall. Conversely, a lower threshold value would produce a lower precision with a higher recall. A curve that has higher precisions among all recall levels implies better in performance. We examined several tens of threshold values.

4.3 Experimental Results

Fig. 5 shows the recall vs. precision for the test sequence *Challenge at Glen Canyon*. It is a 27-minute documentary video including 48,450 frames. It describes the methods employed in repairing the spillway at Glen Canyon Dam. It has 247 shot boundaries consisting of 232 cuts, 11 dissolves and 4 fades. It seems to be made from films by digitizing them, and thus, they include scratch noises, flickers, and position shifts to some extent. *Two-step* in the figure denotes the proposed algorithm. *One-step* algorithm is a part of the two-step algorithm. It is implemented for the purpose of evaluating frame distance-based detection described in Section 3.2. The two-step approach is a device for detecting gradual transitions described in Section 3.3. Every detection algorithm marks satisfactory performance for cut detection as expected. In color code histogram, twin-comparison and one-step algorithms, a few fades causes an explosion of false positives at higher recall levels. In contrast, the two-step detection algorithm is robust for them and marks the best performance.

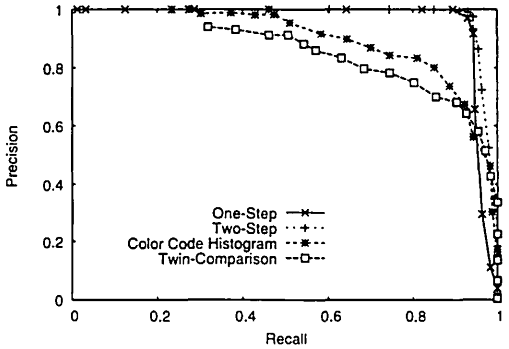

Fig. 5 Precision vs. recall for *Challenge at Glen Canyon*

Fig. 6 shows the recall vs. precision for the test sequence *V*1-1. It is an 47-minute news program including 71,379 frames. It has 477 shot transitions consisting of 373 cuts, 90 dissolves, 7 wipes and 7 special transition effects. Various topics are included such as artist interviews, conferences, sports and art galleries. The proposed two-step detection algorithm keeps high precision over a wide range of recall, while the other detection algorithms decrease the precisions as the recall level increases because of gradual transitions such as dissolves and wipes.

Similar results are obtained for the other test video sequences in our experiments. Two frame-based detection techniques, color code histogram and the one-step algorithm, well detect a cut, however, they are sometimes too sensitive to short gradual transitions, camera motions and noises. On the other hand, they are insensitive to slow dissolves. Twin-comparison and the proposed two-step algorithm improve those frame distance-based algorithms in terms of robustness against noises and sensibility to slow dissolves.

4.4 Current Limitations

Several false positives are caused by camera motions such as zooming and panning. A camera motion produces similar frame distance and interval distance as a gradual transition does. A camera

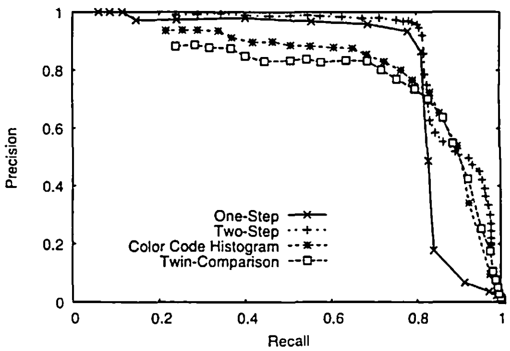

Fig. 6 Precision vs. recall for *V*1-1

motion is differentiated from a gradual transition by performing a motion estimation or an optical flow analysis. Zhang[10] performed a motion analysis after their twin-comparison detection to reduce false positives. A classification of dominant camera motion is a major contribution as well as step-variable in Xiong's paper[11]. It is our future plan to implement camera motion analysis after the proposed two-step detection.

Some test sequences are involved with a fast motion of a large object. It tends to be falsely detected as a shot transition because the frame distance around a large object in motion acts like nothing but a cut. It is hard to distinguish a large moving object from a cut without understanding the shot content. Such a semantic analysis suffers from a time consuming problem.

This paper has not dealt with the determination of a threshold value for shot boundary detection. Instead, a wide range of threshold values is tested for calculating recall and precision values. Some parameters are manually optimized. In practical applications, a methodology for automatic selection of threshold value is helpful. This issue could be solved by several approaches such as in[10] that is based on the distribution of frame distances.

5. Concluding Remarks

We have proposed a two-step algorithm for detecting video shot boundaries. It accurately captures gradual shot transitions as well as abrupt shot transitions. It computes the distance between video intervals and between frames based on the significance map of spatio-temporal wavelet coefficients. Recall-precision curves were computed for several test video sequences. The proposed two-step detection algorithm has shown the best performance among existing two algorithms in our experiments.

The two-step method is capable of detecting video shot boundaries on Motion-JPEG 2000 compressed data since a feature vector can be constructed from its partially decoded bitstream. We are planning to build a content-based video retrieval scheme by using a feature vector defined by Eq.(1).

References

1) R. C. Veltkamp, H. Burkhardt, and H.-P. Kriegel: "State-of-the-Art in Content-Based Image and Video Retrieval", Kluwer Academic Publishers (2001).

2) M. S. Lew: "Principles of Visual Information Retrieval", Springer-Verlag, London (2001).

3) S. F. Chang, T. Sikora, and A. Puri: "Overview of the MPEG-7 standard," IEEE Trans. on Circuits and Systems for Video Technology, Vol. 11, No. 6, pp. 688-695 (2001).

4) S. Hasebe, S. Muramatsu, S. Sasaki, and H. Kikuchi: "Video Querying Based on Three-dimensional Wavelet Transforms", Proc. of ITC/CSCC 2001, pp. 1196-1199 (2001).

5) A. Del Bimbo: Visual Information Retrieval, Morgan Kaufmann Publishers, San Francisco (1999).

6) Y. Yusoff, W. Christmas, and J. Kitter: "A Study on Automatic Shot Change Detection", Proc. 3 rd European Conference on Multimedia Applications, Services and Techniques (ECMAST) (May 1998).

7) S. Hasebe, S. Muramatsu, S. Sasaki, J. Zhou, and H. Kikuchi: "Two-Step Algorithm for Detecting Video Shot Boundaries in a Wavelet Transform Domain", 3 rd International Symposium on Image and Signal Processing and Analysis (ISPA 03), pp. 245-250, Rome, September 18-20 (2003).

8) J. S. Boreczky and L. A. Rowe: "Comparison of video shot boundary detection techniques", Proc. SPIE, 2670, pp. 170-179, San Diego (1996).

9) U. Gargi, R. Katsuri, and S. H. Strayer: "Performance Characterization of Video-Shot-Change Detection Methods", IEEE Trans. Circuits and System for Video Technology, Vol. 10, No. 1, pp. 1-13 (2000).

10) H. J. Zhang, A. Kankanhalli, and S. W. Smoliar: "Automatic Partitioning of Full-Motion Video", ACM Multimedia Systems Journal, pp. 10-28 (1993).

11) W. Xiong and J. C. M. Lee: "Efficient Scene Change Detection and Camera Motion Annotation for Video Classification", Computer Vision and Image Understanding, Vol. 71, Issue 2, pp. 166-181 (1998).

12) W. Xiong and J. C. M. Lee: "Net Comparison: A Fast and Effective Method for Classifying Image Sequences", Proc. SPIE Storage and Retrieval for Image and Video Database III, Vol. 2422, pp. 313-328, San Jose (1995).

13) J. Nang, S. Hong, and Y. Ihm: "An Efficient Video Segmentation Scheme for MPEG Video Stream Using Macroblock Information", Proc. 7 th ACM International Conference on Multimedia, pp. 23-26 Orlando (1999).

14) S. B. Jun, K. Yoon, and H. Y. Lee: "Dissolve Transition Detection algorithm Using Spatio-Temporal Distribution of MPEG Macro-Block Types", Proc. 8 th ACM International Conference on Multimedia, pp. 391-394, Mariana del Rey (2000).

15) J. Bescos: "Real-Time Shot Change Detection Over Online MPEG-2 Video", IEEE Trans. on Circuits and Systems for Video Technology, Vol. 14, Issue 4, pp. 475-484 (2004).

16) T. Fukuhara, K. Katoh, S. Kimura, K. Hosaka, and A. Leung: "Motion-JPEG 2000 standardization and target market", Proc. IEEE ICIP,No. TA 0208, Vancouver (2000).

17) D. S. Taubman and M. W. Marcellin: JPEG 2000 Image Compression Fundamentals, Standards and Practice, Kluwer Academic Publishers, Massachusetts (2002).

18) C. E. Jacobs, A. Finkelstein, and D. H. Salesin: "Fast multiresolution image querying", Proc. SIGGRAPH'95, ACM, pp. 227-286 (1995).

19) X. Wen, T. D. Huffmire, H. H. Hu, and A. Finkelstein:

"Wavelet-based video indexing and querying", Multimedia Systems, Vol. 7, Issue 5, pp. 350-358, Springer Verlag Heidelberg (1999).

20) J. Meng, Y. Juan, and S. F. Chang: "Scene Change Detection in a MPEG Compressed Video Sequence", Proc. IS & T/SPIE Symp., Vol. 2419, pp. 14-25 (1995).

21) W. Sweldens and P. Schröder: "Building your own wavelets at home", SIAM J. Math., 29, 2, pp. 511-546 (1997).

22) I. Daubechies and W. Sweldens: "Factoring wavelet transforms into lifting steps", J. Fourier Anal. Appl., Vol. 4, No. 3, pp. 245-267 (1998).

23) E. J. Stollnists, T. D. Derose, and D. H. Salesin: Wavelets for Computer Graphics, Morgan Kaufmann Publishers, San Francisco (1996).

24) V. Castelli and L. D. Bergman: Image Databases, Jhon Wiley & Sons (2002).

25) The Open Video Project, http: //www.open-video.org/.

26) ISO/IEC JTC 1/SC 29/WG 11, Licensing Agreement for the MPEG-7 Content Set, Atlantic City (1998).

(Received April 30, 2004)

Satoshi Hasebe

Received B. E. and M. E. degrees from Niigata University, Niigata, in 2000 and 2002, respectively. He is currently a candidate for the D. E. degree at Niigata University. His research interests include image/video analysis, processing and indexing.

Makoto Nagumo

Received B. E. and M. E. degrees from Niigata University, Niigata, in 2002 and 2004, respectively. He joined Broadcasting System of Niigata, Inc. (BSN) in 2004. Currently, he is involved in terrestrial digital broadcasting systems as well as in radio and television broadcasting operations and transmissions.

Shogo Muramatsu

Received B. E., M. E., and D. E. degrees in electrical engineering from Tokyo Metropolitan University in 1993, 1995, and 1998, respectively. From 1997 to 1999, he worked at Tokyo Metropolitan University. In 1999, he joined Niigata University, where he is currently an associate professor at Department of Electrical and Electronic Engineering. During a year from 2003 to 2004, he was a visiting scientist at University of Florence, Italy. His research interests are in digital signal processing, multirate systems, image processing and VLSI architecture. Dr. Muramatsu is a member of IEEE (Institute of Electrical and Electronics Engineers, Inc.), IPSJ (Information Processing Society of Japan) and the IEICE (Institute of Electronics, Information and Communication Engineers).

Hisakazu Kikuchi (Member)

Received his B. E. and M. E. degrees from Niigata University, Japan, in 1974 and 1976, respectively, and Dr. Eng. degree in electrical and electronic engineering from Tokyo Institute of Technology, Japan, in 1988. From 1976 to 1979 he worked at Information Processing Systems Laboratory, Fujitsu Ltd., Tokyo. Since 1979 he has been with Niigata University, where he is a professor of electrical engineering. He was a visiting professor at Electrical Engineering Department, University of California, Los Angeles during a year of 1992 to 1993. He holds a visiting professorship at Chongqing University of Posts and Telecommunications, China since 2002. His research interests are in digital signal processing and image/video processing. Dr. Kikuchi is a member of IIEEJ, IEICE, ITE, Japan Society for Industrial and Applied Mathematics, Research Institute of Signal Processing, IEEE, and SPIE. He served the chair of Circuits and Systems Group, IEICE, in 2000 and the general chair of Digital Signal Processing Symposium, IEICE, in 1988 and Karuizawa Workshop on Circuits and Systems, IEICE, in 1996.