MVA2005 IAPR Conference on Machine VIsion Applications, May 16-18, 2005 Tsukuba Science City, Japan
3-32

Scene change detection using Multi-class Support Vector Machine with MPEG encoding information

Mickael Pic and Takio Kurita
Neuroscience Research Institute
National Institute of Advanced Industrial Science and Technology
Tsukuba AIST Central 2, Umezono 1-1-1, Tsukuba 305-8561, Japan

1 Introduction

As the amount of digital video is increasing, efficient ways of searching and annotating it according to its content are required. The first step toward video indexing is to detect scene changes. A scene is usually defined as a sequence of video frames with no significant changes between frames in terms of their visual content. The simplest scene cut is represented by a camera break, that is, an abrupt transition due to an editing cut. More sophisticated changes are gradual transitions such as dissolves, wipes, fade-ins, fade-outs, resulting from chromatic, spatial, and combined edits.

In this paper we address the problem of abrupt transition recognition or, briefly, video-cut detection. Most techniques of video segmentation works on uncompressed data and rely on features such as color histograms ([1]), tracking of feature points ([2]), subsequent frame differences ([3]), and motion features ([4]). Some algorithms have also been developed to work directly on MPEG-encoded video sequences ([5][6]), and have improved the computational efficiency, video compression is also generally done with signal-processing techniques capable of deriving useful features, for example, motion vectors in MPEG.

While techniques that work in the uncompressed domain usually achieve high robustness, techniques that work directly on MPEG-encoded video are usually faster. We propose an original method combining the robustness of the uncompressed domain to the speed of MPEG-encoded video and present the first experimental results. Almost all these methods rely on a threshold selected by a human operator. Because it is not always easy to manually find a good threshold when several features are used, the operator uses a near optimal threshold. Sometimes, neural networks ([7]) are used to determine these thresholds. While they can better find relations between the features, they can be slow to train. We are proposing an algorithm that can extract seven features from MPEG-encoded information, three features from DC components and four features from B-frame macroblocks. We use these features to train a Multi-class Support Vector Machine ([8]) to automatically design a classifier for detecting video cuts.

The organization of the paper is as follows: Section 2 provides a brief description of the MPEG video-compression standard, while Section 3 describes details on the proposed algorithm to detect cuts. Section 4 presents the results of an experimental study.

2 MPEG Video Encoding

To begin, we will briefly describe the relevant parts of the MPEG video-compression standard ([9]). The syntax for MPEG video defines three main types of coded pictures organized into sequences of groups of pictures (GOP) in MPEG video streams. The Intra-coded pictures (I-frames) are obtained without exploitating the temporal redundancy representing the reference frame for other frames in the GOP. The Predicted pictures (P-frames) are forward-motion-compensated encoded pictures, starting from the previous I- or P-frame. And the Bi-directionally predicted pictures (B-frames) are forward- and backward-motion-compensated encoded pictures, starting from the previous or following I or P frames. Figure 1 outlines a typical GOP structure (15-frame sequence of IBBPBBPBBPBBPBB) that is used in coding video at a rate of 30 frames/sec.

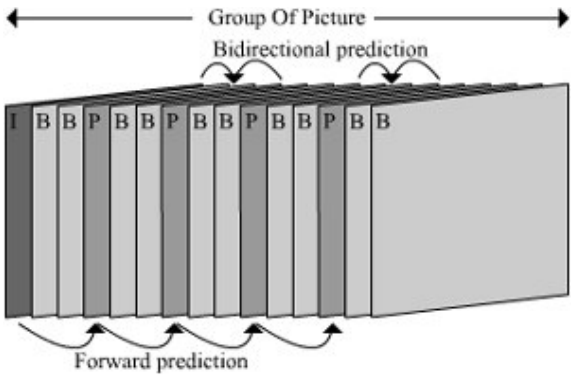

Figure 1: Typical MPEG group of pictures (GOP)

A video frame is divided into a sequence of non-overlapping macroblocks. Each macroblock consist of six 8 x 8 pixel blocks, four luminance (Y) blocks, and two chrominance (CbCr) blocks. Each macroblock is intra- or inter-coded. An I-frame is completely intra-coded.

Since the coding for an I-frame does not refer to any other video frames, it can be decoded independently and thus provides an entry point for fast random access to the compressed video. Each P frame

is predictively encoded with reference to its previous anchor frame, the previous I- or P-frame. Each macroblock in the P-frame, is search for a local region in the anchor frame that is a good match in terms of the difference in intensity. If a good match is found, the macroblock is represented by a motion vector to the position of the match. This is normally known as encoding with *forward motion compensation* and we shall refer to this type of encoded macroblock as an *inter-coded macroblock*. If a good match cannot be found, the macroblock is intra-coded like in the I-frames. An inter-coded macroblock also has better compression gain compared to intra-coded macroblocks. We expect that a small change in content between a P frame and its anchor frame will result in more well matched macroblocks in the anchor frame and hence fewer macroblocks requiring intra-coding. To achieve further compression, B-frames are Bi-directionally predictively encoded with forward and/or backward motion compensation references for their nearest past and/or future I- and/or P-frames. Since B-frames are not used as a reference for coding other frames, they can accommodate more distortion and thus higher compression gain compared to I- or P-frames.

3 Cut Detection Algorithm

The method we propose sequentially reads the encoded information from each B-frame of a MPEG movie. This information is preprocessed before being used in a Multi-class Support Vector Machine ([8]). The MSVM gives the final category for the frame.

3.1 MPEG-encoded information preprocessing

Our algorithm only uses information from the B-frames because they achieve the highest compression and they convey more useful information than I- and P- frames. Junehwa and Boon-Lock ([10]), described a method of extracting a small representation of the full frame from compressed information using DC coefficients, not requiring the frame to be fully uncompressed, thus saving a great deal of computational time. This compressed frame is called the *DC image*, and is represented in the YUV color space. In our algorithm, we extract three features from consecutive *DC images*.

The measure of difference between consecutive *DC images* is important in detecting cuts. Mean squared error (MSE) is often used as the measure of difference between two images. However, MSE is affected by objects moving in the frame or the abrupt appearance of captions. To reduce the influence these intrusions have, we used the robust mean squared error (RMSE) [11] in our algorithm. Suppose the Y components of consecutive *DC images* are denoted as $Y^{(1)} = (y_1^{(1)}, \ldots, y_M^{(1)})$ and $Y^{(2)} = (y_1^{(2)}, \ldots, y_M^{(2)})$. Then the mean squared error is defined by

$$\text{MSE} = \frac{1}{M}\sum_{i=1}^{M}(y_i^{(1)} - y_i^{(2)})^2 = \frac{1}{M}\sum_{i=1}^{M}\varepsilon_i^2. \qquad (1)$$

In the robust mean squared error (RMSE), outliers (pixels corresponding to moving objects) are detected by accurately estimating the standard deviation and the mean squared error is computed only with inliers. Outliers between the consecutive Y components of *DC images* $Y^{(1)}$ and $Y^{(2)}$ can be detected with the following procedure[11]:

1. Compute the median of square errors $\varepsilon^2 = \text{med}\,\varepsilon_i^2$.
2. Compute the estimated standard deviation as $\sigma = 1.4826(1 + \frac{5}{M-1})\sqrt{\varepsilon^2}$.
3. Determine pixels i as outliers if $\sqrt{\varepsilon_i^2} \geq 2.5\sigma$.

Then, the robust mean squared error of the Y component is obtained by calculating the mean squared error of the inliers. We have called this feature RMSEY.

Color information is also important in measuring the difference between consecutive images. The second and third features are defined using the U and V components of consecutive *DC images*. To extract the U feature (CHU) and V feature (CHV), the sum of the absolute difference in values in corresponding bins of histograms is used. This is also called *bin to bin difference* ([1]). Given two histograms h_1 and h_2

$$fd_{b2b} = \frac{1}{2N}\sum_i abs(h_1[i] - h_2[i]) \qquad (2)$$

where N is the number of pixels in a frame and factor 2 ensures that even for completely non-intersecting histograms, the difference in frames is less than or equal to 1.0.

In addition to the features extracted from *DC images*, four more features are extracted from B frame macroblocks. When a movie is encoded in MPEG format, B frames macroblocks can be coded using four different types ([9]). For each macroblock, the encoder calculates the best motion-compensated macroblock for forward-motion compensation (FWD). It then calculates the best motion compensated macroblock for backward motion compensation (BWD) with a similar method. Finally, it averages the two motion-compensated macroblocks to produce the interpolated macroblock (IP). It then selects the one with the best performance. After this step, if the motion compensated macroblock is only slightly better than the uncompensated macroblock, then the macroblock is coded in intra-mode (IA). While reading the macroblock information from the movie, the system counts how many macroblocks belong to each class.

The features RMSEY, CHU, and CHV are computed for every frame of the movie, because we need to compare the current B frame and previous I-, P- or B-frame color information. The features FWD, BWD, IP, and IA on the other hand are only computed for B frames.

3.2 Multi-class Support Vector Machine

We propose an algorithm for detecting shot boundaries with the Multi-class Support Vector Machine MSVM ([8]). The MSVM is an extension of the original binary classification Support Vector Machine SVM algorithm.

Because we are only analyzing B frames, and cuts may occur on any frame (I, P or B), we defined three B-frame type classes: No Cut frames (NC), Before Cut frames (BC), and On Cut frames (OC). The NC occurs when there is no cut on the current B-frame, or the next frame (I, P or B). The BC occurs when-ever a cut appear on the next frame, or the second-next frame when the cut is on a I- or P-frame, following the current frame. For example, in the frame sequence BBP, if a cut occurs on the P-frame, the previous two B frames will be labeled BC. The OC occurs when the current frame is a cut frame (first frame of a new sequence). We do not define an After Cut frame because very often the MPEG-encoding information indicates an important change just before a cut happens, and less often after a cut.

The classifier was trained using a polynomial Kernel

$$k(\boldsymbol{x}, \boldsymbol{x}') = (\gamma < \boldsymbol{x}, \boldsymbol{x}' > +c)^2 \qquad (3)$$

and sub-sampled features of full video sequences including scene cuts, and camera motion (the videos are described in Section 4). We trained several MSVM with different parameter settings to find the near optimal setting.

By analyzing the results of the classifier, the algorithm can determine the cut-frame. In the following MPEG sequence: PBBPBB, three situations can occur. There is no cut, and the classifier will give the following result (only B-frames are analyzed) - NC NC - NC NC. There is a cut on one of the B-frames - BC OC - NC NC. The BC indicate that the following frame is a cut, and this is confirmed by the OC. There is a cut on a non B-frame - BC BC - NC NC. In that case, usually the previous two B-frames are classified as BC. Only the following non B-frame is set as the cut-frame.

4 Experimental result

Numerous shot-boundaries detection algorithms have been proposed and evaluated using different video sequences making it difficult to compare their performance. We captured three long videos for our experiments in MPEG2 320 x 240 pixel format at 30 frames per second, from television sources. Table 1 lists the information from the three videos used in the experiments. All videos include fast camera motion (panning, tilting, zooming), flash and special effects. They also have several kinds of scene cuts, from a simple camera brake to sophisticated fading and wiping. As we focused on camera breaks, we ruled out sophisticated scene cuts from the experiments. News_1 and News_2 were TV news, with commercial breaks. Kao_06-10 was a TV drama with commercial breaks. All frames for all movies were manually labeled according to the three classes used by the classifiers: No Cut frames, Before Cut frames, and On Cut frames. Because the proposed algorithm works directly on the MPEG compressed domain, it can achieve real-time performance. In most papers, experiments are done on very short movie, a few thousand frames or seconds at most, and few scene cuts or camera motions. To show the effectiveness of our method, we choose to use a very long video with a lot of scene cuts, and camera movements.

Table 1. Video information

Video	*Frames*	*Shot boundaries*	*Duration*
News_1	26,895	105	14 mins
News_2	17,361	58	9 mins
Kao_06-10	102,323	991	52 mins

The MSVM was trained using 13,400 frames subsampled from News_1 and News_2 including scene breaks, camera motion, and flash.

Many algorithms have been proposed to detect scene cuts, using different video sequences for evaluation. To provide a comparison, we tested three well-known algorithms against three of our own methods using the videos we captured. For the well-known algorithms, we programmed the MSE as defined by Eq 1, a Color Histogram (CH) ([1]) with *bin to bin difference* as defined by Eq 2, and Rapid Scene Analysis on Compressed Video (RSACV) ([5]). For our own methods, we programmed the RMSE defined in Section 3.1, a simply modified Categorization And Regression Tree (CART) ([12]), and the MSVM.

Prior to use MSVM as classifier for our proposed algorithm, we tried a simpler classifier Classification And Regression Tree CART as presented by Breiman et al.[12] to the problem of detecting shot boundaries using the seven features extracted from the MPEG compressed video. Using the same terminology for labeling the frames as MSVM uses, CART was trained to classify the frames as No Cut frames (NC), Before Cut frames (BC), and On Cut frames (OC). We kept CART to show the differences between the two classifiers.

To improve the ability to generalize of the constructed decision tree, we introduced a small modification to the CART training algorithm. When the best split of a node has been chosen while growing a tree, the data that belong to the feature chosen as the split are sorted, and the value of the split is taken as the middle value between the best split and the next value. This makes the tree robust against small variations in feature value that can appear in the test samples and not in the training samples.

The CART classifier was trained using the same data set as the MSVM classifier.

We present the results of each feature taken separately, the result of RSACV, our first method using a

modified CART, and our new proposed method using MSVM in the following tables.

Table 2: Experimental results for News_1

Method	*Miss detect*	*False detect*	*Total*
MSE	10	47	57
CH	4	39	43
RSACV	10	51	61
RMSE	14	20	34
CART	0	0	0
MSVM	2	8	10

Table 3: Experimental results for News_2

Method	*Miss detect*	*False detect*	*Total*
MSE	14	63	77
CH	8	34	42
RSACV	2	55	57
RMSE	8	1	9
CART	0	1	1
MSVM	4	4	8

We can see from Table 4 that the proposed method achieved better results than the other methods, and features taken separately. Yet in Table 2 and Table 3 CART appear to achieve better result than MSVM. As stated before, the training set has been created from News_1 and News_2 samples. So testing with News_1 and News_2 can be consider as recall test, and testing with Kao_06-10 is a generalization test. The improvement on Kao_06-10 derives from the MSVM's ability to automatically combine features to separate classes. The proposed modification to the CART to select the value for the best split at each node of the tree makes the constructed tree robust against variations in feature values that do not appear in the training samples. While the recall process yields better results for the CART, the MSVM achieves better generalization. Even if CART appear to have similar performance than MSVM, our goal is to create a classifier than can learn from examples to generalize and achieve good results at shot boundaries detection.

Table 4: Experimental results for Kao_06-10

Method	*Miss detect*	*False detect*	*Total*
MSE	110	114	224
CH	88	283	371
RSACV	59	39	98
RMSE	59	90	149
CART	44	58	102
MSVM	38	48	86

References

[1] R Katsuri, S H Strayer, U Gargi, and S Antani. An evaluation of color histogram based methods in video indexing. *Proc. of Int. Workshop on Image Database and Multi-media Search*, 1996.

[2] Y Abdeljaoued, T Ebrahimi, C Christopoulos, and I Mas Ivar. A new algorithm for shot boundary detection. *Proc. European Signal Processing Conference, Tampere*, pages 151–154, 1998.

[3] M Yazdi and A Zaccarin. Scene break detection and classification using a block-wise difference method. *ICIP*, 2001.

[4] B Erol and F Kossentini. Partitioning of video objects into temporal segments using local motion information. *Proc. ICIP*, 2000.

[5] Boon-Lock Yeo and Bede Liu. Rapid scene analysis on compressed video. *IEEE Transactions on Circuits and Systems for Video Technology*, 5(6), December 1995.

[6] G Akrivas, N D Doulamis, A D Doulamis, and S D Kollias. Scene detection methods for mpeg – encoded video signals. *Proc. MELECON 2000 Mediterranean Electrotechnical Conference, Nicosia, Cyprus*, May 2000.

[7] E Ardizzone, G A M Gioiello, M LaCascia, and D Molinelli. A real time neural approach to scene cut detection. *IS&T/SPIE Storage and Retrieval for Media Database, San Jose, California*, 2670, 1996.

[8] C.-W Hsu and C.-J Lin. A comparison on methods for multi-class support vector machine. *IEEE Transaction on Neural Networks*, 13:415–425, 2002.

[9] ISO. Information technology – coding of moving pictures and associated audio signal for digital storage median at up to about 1,5 mbit/s – part 2: Video. *ISO/IEC 11172-2*, 1993.

[10] S Junehwa and Y Boon-Lock. Fast extraction of spatially reduced image sequences from mpeg-2 compressed video. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(7), 1999.

[11] T Kurita. Robust template matching and its application to cut detection. *Proc of the 1997 IEICE General Conference*, D-12-61:268, 1997.

[12] L Breiman, J H Friedman, R A Olshen, and C J Stone. Classification and regression trees. *Wadsworth & Brooks*, 1984.