SUPERVISED CLASSIFICATION FOR VIDEO SHOT SEGMENTATION

Yanjun Qi, Alexander Hauptmann, Ting Liu

School Of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213

ABSTRACT

In this paper, we explore supervised classification methods for video shot segmentation. We transform the temporal segmentation problem into a multi-class categorization issue. This approach provides a uniform framework for using different kinds of features extracted from the video and for detecting various types of shot boundaries. The approach utilizes manual labeled training data and a simple classification structure, which eliminates arbitrary thresholds and achieves more reliable estimation than previous threshold-based methods. Contrastive experiments on 13 videos (~4 hours) show excellent performance on the 2001 TREC Video Track Shot Classification Task in terms of precision and recall.

1. INTRODUCTION

Temporal video segmentation is the first step toward automatic annotation of digital video for browsing and retrieval. Its goal is to divide the video stream into a set of meaningful and manageable segments called shots. A shot is defined as an unbroken sequence of frames taken from one camera. There are two basic types of shot transitions: cut and gradual. A gradual transition may occur in many forms, with fades and dissolves the most frequent. Gradual transitions are more difficult to detect than cuts. They must be distinguished from camera operations and object movements.

We consider video shot segmentation from a different perspective, namely as a categorization task, classifying every frame in the video stream as either a "common shot frame", a "cut frame", a "fade frame", or a "dissolve frame". Adjacent frames tend to be in the same class and usually have only small differences between them. This classification framework allows us to use many different kinds of video features in an integrated structure. The supervised learning process also enables reliable estimation of thresholds, which has not been addressed by other shot segmentation research so far.

This paper is organized as follows. The next section reviews the related work. Section 3 presents the basic features, classification strategies and process of this supervised shot segmentation. Section 4 gives experimental results, and we conclude with a summary.

2. RELATED WORK

Good overviews of existing techniques in temporal video segmentation operating on both uncompressed and compressed video streams are found in [1][2][5].

For uncompressed data, basically, most algorithms are based on frame differences for pixel, block-based or histogram comparisons. Most existing methods rely on suitable thresholding of differences between successive frames. However, these thresholds are typically highly sensitive to the specific type of video. There have only been a few machine learning approaches that tried to overcome this drawback. [9] views temporal video segmentation as a 2-class clustering problem ("scene change" and "no scene change") and uses K-means to cluster frame differences. [4] applies HMMs with separate states to model shot cuts, fades, dissolves, pans and zooms. [2] proposes a reliable dissolve detector. They use a 'dissolve synthesizer' to create an infinite amount of dissolve examples of arbitrary duration, as artificial training data for supervised learning methods. [5] provides a statistical detector based on minimization of the average detection-error probability for cuts and dissolves.

As pioneered by the above methods, classification methods appear promising for this task. However, most existing shot detection algorithms just use ad hoc frame classification with arbitrary thresholding rules. In the following section, we present a reliable shot boundary detection approach based on supervised classification and later validate its usefulness.

3. SHOT SEGMETNATION BASED ON SUPERVISED CLASSIFICAITON

3.1. Problem Analysis and System Overview

We treat every frame in the video stream as a single feature vector and classify each frame into exactly one class. Although the supervised classification is capable of detecting many different types of shot boundaries, we demonstrate its performance on just two broad boundary classes in this paper: hard cuts and gradual transitions. The

0-7803-7965-9/03/$17.00 ©2003 IEEE

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR. Downloaded on June 03,2020 at 13:47:02 UTC from IEEE Xplore. Restrictions apply.

classification works in a two level hierarchy. The dashed block in Figure 1 shows that, first, cut frames are distinguished from non-cuts, and then non-cuts are split into gradual transition frames and normal frames. The system uses features including frame differences, as well as camera motion likelihood, and black frame likelihood.

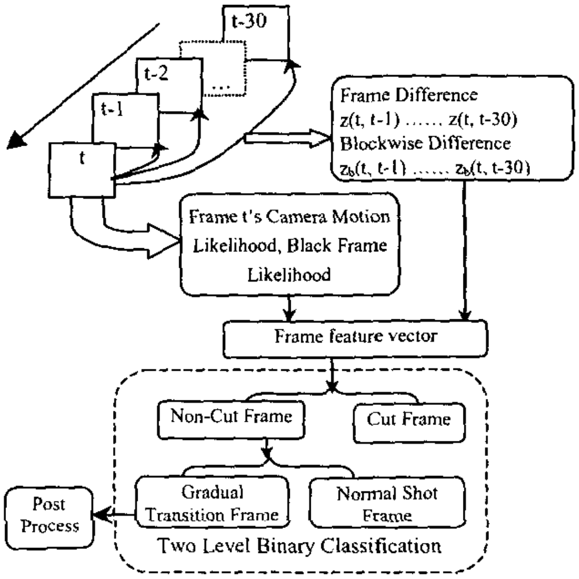
t-30
t-2
t-1
t
Frame Difference
z(t, t-1) ...... z(t, t-30)
Blockwise Difference
z_b(t, t-1) ...... z_b(t, t-30)
Frame t's Camera Motion Likelihood, Black Frame Likelihood
Frame feature vector
Non-Cut Frame
Cut Frame
Gradual Transition Frame
Normal Shot Frame
Post Process
Two Level Binary Classification

Figure 1. System Overview

The following subsections describe each process in detail.

3.2. Deriving the Vector of Frame Features

To categorize each frame into three different classes, we need to use features that reliably distinguish the different segmentation categories. Our features can be divided into two types: frame differences and current frame properties.

3.2.1. Frame Difference

Frame differences show different behavior for frames within shots, at cut boundaries or around gradual transitions. Within a shot, the differences between frames can mainly be attributed by object/camera motion or lighting changes. Thus, desirable features for shot classification will be insensitive to motion and lighting.

To compute frame differences, we extract features that represent the visual content of a frame. Then, we quantify the difference between frame k and frame $k+1$. Two sets of frame differences are used: A whole-frame color histogram difference and an 8*8 block-wise histogram difference between frames, both in the YUV color space.

Histogram differences are insensitive to object motion and invariant to image rotation. Experiments in [9] show that histogram differences in YUV color space achieve the best overall performance among different color spaces. [5] reports that a block-wise histogram difference is very sensitive to cut boundaries, but, because of the emphasis on blocks, it is also sensitive to object and camera motion. Combining both global and block wise histogram differences makes the system insensitive to object motion but very sensitive to cut boundaries.

For each frame, we calculate the feature differences for each of 30 frame pairs between frame t and frame $t-1$, up to frame t and frame $t-30$, for both the global and the 8*8 block wise histogram features. These 60 window-based differences represent a frame's temporal relationship with its neighborhood. Although there is no absolute minimum duration for a shot, it can realistically be assumed that no shot last for less than a second. Thus, the 30-frame difference window is close to the minimum shot length at 30 frames per second.

3.2.2. Current Frame Property

Further improvement of the detection performance can be achieved by using extra information. Although the histogram difference is insensitive to object motion, it remains somewhat sensitive to camera motion, such as panning, or zooming. Therefore, we added one feature containing the likelihood of a camera motion at the current frame. This value is the sum of the probabilities for zoom in/out as well as pan left/right/up/down camera actions. These are derived from an analysis of the motion vectors in the MPEG-1 compressed stream [10].

[1] points out that none of the difference measures perform satisfactorily on very dark frames. Very black frames frequently appear during fade transitions, which constitute one kind of gradual transition. Thus we added another feature representing the likelihood that the current frame is black. This is merely the average Y value of the frame.

3.3. Hierarchical Classification

As illustrated in Figure 1, after we build a feature vector for each frame, we first use a binary classifier to categorize the frames into "non-cut frames" or "cut frames". For the "non-cut frames", we then use the second level binary classifier to distinguish a "shot frame" from a "gradual transition frame". In general, distinguishing cuts from gradual transitions or normal shots is easier than separating gradual frames from normal shot frames. We compared the following classifiers for binary classification in our system:

KNN. KNN stands for k-nearest neighbor classification. The algorithm is quite simple: given a feature vector, the system finds the k-nearest neighbors among the training vectors, and uses the categories of the k neighbors to determine the category of this test vector. For the binary case used in this paper, for each test frame, we calculate the ratio of positive examples within its k nearest neighbors and used this value as the positive likelihood.

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR. Downloaded on June 03,2020 at 13:47:02 UTC from IEEE Xplore. Restrictions apply.

NB: The basic idea in Naïve Bayes (NB) probabilistic classification is to use the features' joint probabilities to estimate the probabilities of a category given a data point. The Naïve Bayes assumption of feature independence makes the computation of the NB classifiers far more efficient than the non-naïve Bayes approaches.

SVM: Support Vector Machines (SVM) is a relatively recent but increasingly popular learning approach for solving two-class pattern recognition problems. It is based on the structural risk minimization principle. The method aims to find a decision surface that "best" separates the data points in two classes. The SVM formulation can be solved using quadratic programming techniques.

3.4. Post Processing for Gradual Transitions

Due to the variations in video stream, the classification score for each frame is also quite noisy. This is not a big problem for the first level classification, because cut frames are usually easy to distinguish. But the noise may cause errors in the detection of a gradual transition, which extends over several adjacent frames.

We use wavelet smoothing to perform an automatic de-noising process on each non-cut frame's second level classification score. This wavelet smoothing helps to erase the noise and consolidate the classification scores corresponding to a sequence of gradual transition frames.

Additionally, we take the presence or absence of a nearby shot boundary into account when we perform the temporal integration for graduals, because multiple transitions are unlikely to be immediately adjacent to each other.

4. EXPERIMENTS

4.1. TREC-2001 Video Data Set

There has been a long time need for standard benchmarks and unified evaluation criteria in shot boundary research. The TREC-2001 Video Track [6][7][8] organized by NIST provided such a data corpus that allows consistent comparison and precise evaluation of different systems. The video collection consists mostly of documentary style videos, widely varying in age, production style and quality. This corpus contains about 5.8 hours MPEG-1 videos. Discarding some of the extremely short videos, we used about 4 hours of video from this corpus, or 13 MPEG-1 video files at slightly over 2GB of data. This collection contained 420,976 frames and 2462 transitions, of those 1670 were cuts (67%), and 792 gradual transitions.

4.2. Evaluation

For all videos in this corpus, shot segmentation reference data had been constructed manually by NIST. We compared our detection results for each video to the shot segmentation reference data using evaluation software provided by NIST to get performance measurements for each video. We report the Precision/Recall score to evaluate our experiments.

Precision: Among the transitions (cut or gradual) detected by the system, how many are true transitions?

Recall: For all possible transitions (cut or gradual), how many were detected by system?

A good detector should have both high precision and high recall. If we let d be the number of transitions detected, z be the number of transitions manually judged, and dz be the number of transitions manually judged as transitions among the detected ones, the precision and recall for the system are:

$$\text{Precision} = dz/d \quad \text{Recall} = dz/z$$

F1 is a commonly used metric that combines precision and recall.

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

F1 is high only when precision and recall are both high.

4.3. Experimental Results

While we conducted numerous experiments, we present here four experimental runs that provide interesting and meaningful contrasts:

Run-1 (30.bc.bc): This run uses only block wise histogram difference (30 features) and NB for both levels of classification.

Run-2 (30.knn.knn): Uses block wise histogram difference (30 features) and kNN for both levels of classification.

Run-3 (62.knn.knn): Uses both global and block wise histogram differences, camera motion likelihood and black-frame likelihood (30+30+2 features) with kNN for both levels of classification.

Run-4 (62.svm.knn): Use the same 62 features as Run 3. Uses a linear SVM for the first level classification, and kNN for the second level.

In each run, for each video, we trained the classification model on the other 12 videos.

For the kNN classification step, we set two parameters, k and a score cutoff based on one validation video outside the 13 videos in the collection,

From the TREC evaluation software [6], we obtained each test video's cuts/gradual detection precision/recall scores as well as a weighted sum of precision and recall, weighted by the number of transitions in each video.

In addition to comparing the four runs described above, we also compared our results to the best performing systems at the 2001 TREC evaluation [6]. One system was best at cut detection, though not for gradual transitions, while the other system performed best for gradual transitions, but not for cuts. Figure 2 shows the precision vs. recall plots of these six runs. Each run has two points

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR. Downloaded on June 03,2020 at 13:47:02 UTC from IEEE Xplore. Restrictions apply.

in the figure. One is for detection of cuts, the other is for gradual transition detection.

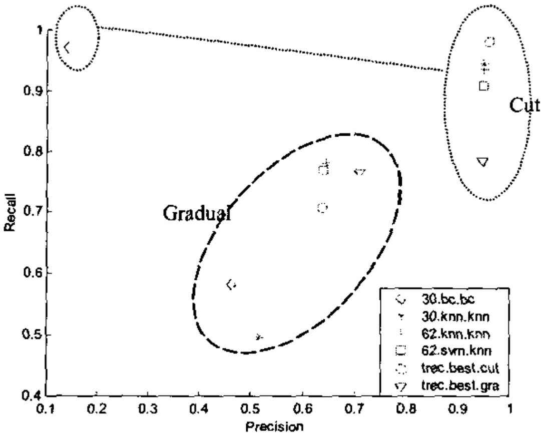

Figure 2. Precision vs. Recall for Cuts and Gradual Transitions

From Figure 2, we can see that for the cut detection, block-wise histogram difference using 30 features (star '*' symbol) is similar to the complete 62-feature set. This confirms the hypothesis that block-based histogram difference is very appropriate for cut detection. But for gradual transition detection, the 62-feature set shows dramatically improved performance (plus '+' symbol). Linear SVM (square symbol) performs similar to kNN ('+' symbol) for cuts. Other, non-linear SVM kernels did not improve performance. In general, NB performs quite poorly here due to our window-wise feature generation.

Table 1 lists the F1 comparison for these six runs. We can see that though our 62.knn.knn run has neither the best in cuts nor in gradual segmentation, but it achieves the overall best performance when both are considered.

RUNS	CUT_F1	GRADUAL_F1	SUM_F1
30.bc.bc	0.241644	0.500100	0.3709
30.knn.knn	0.947389	0.485034	0.7162
62.knn.knn	0.942435	0.698285	**0.8204**
62.svm.knn	0.928222	0.685770	0.8070
TrecBestCut	**0.965900**	0.670600	0.8182
TrecBestGra	0.857200	**0.729700**	0.7934

Table 1. F1 Comparison. The best performing system in each category is in bold type.

5. CONCLUSION

This paper considered video shot segmentation from a supervised classification perspective. It provided a generic technique that provides reliable shot boundary categorization allowing multiple features to be used simultaneously to improve performance. Experiments show that this method has excellent performance on the 2001 TREC Video Track Shot Classification Task in terms of precision and recall.

In the future, it may be productive to add audio features to the classification, as well as exploring other machine learning techniques.

6. ACKNOWLEDGEMENTS

This work was supported in part by the Advanced Research and Development Activity (ARDA) under contract number MDA908-00-C-0037 and by the National Science Foundation under Agreement No.IIS-0105219 and Agreement No. IIS-0121641.

7. REFERENCES

[1] I. Koprinska, S. Carrato, "Temporal video segmentation: a survey," *Signal Processing: Image Communication*, vol. 16, pp. 477--500, 2001

[2] R. Lienhart, "Reliable Transition Detection in Videos: A Survey and Practitioner's Guide", *International Journal of Image and Graphics (IJIG)*, Vol.1, No.3, pp.469-486, 2001.

[3] R. Lienhart and A. Zaccarin, "A system for reliable dissolve detection in Videos", *Proceedings of ICIP 2001*, pp. 406-409 vol.3.

[4] J. S. Boreczky and L. D. Wilcox. "A Hidden Markov Model Frame Work for Video Segmentation Using Audio and Image Features", *Proceedings of ICASSP'98, pp.3741-3744, Seattle*, May 1998.

[5] Alan Hanjalic, "Shot Boundary Detection: Unraveled and Resolved", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 12, No. 2, pp. 90 -105, Feb. 2002.

[6] A.F. Smeaton, P. Over, R. Taban, "The TREC-2001 Video Track Framework", *Proceedings of the Tenth Text Retrieval Conference (TREC-2001), Gaithersburg, Maryland*, November 13-16, 2001

[7] J.R. Smith, etc, "Integrating Features, Models, and Semantics for TREC Video Retrieval", *Proceedings of TREC-2001, Gaithersburg, Maryland*, November 2001.

[8] G.M. Quenot, "TREC-10 Shot Boundary Detection Task:CLIPS System Description and Evaluation", *Proceedings of TREC-2001, Gaithersburg, Maryland*, November 2001.

[9] B.Gunsel, A.Ferman, etc, "Temporal Video Segmentation Using Unsupervised Clustering and Semantic Object Tracking", *Journal of Electronic Imaging* 1998, pp. 592-604

[10] R. Jin, Y. Qi, A. Hauptmann, "A Probabilistic Model for Camera Zoom Detection", *Proceedings of ICPR 2002*, Quebec, August 2002.

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR. Downloaded on June 03,2020 at 13:47:02 UTC from IEEE Xplore. Restrictions apply.