論文 / 著書情報
Article / Book Information

Title(English)	Automatically Estimating Number of Scenes for Rushes Summarization
Authors(English)	Koji Yamasaki, Koichi Shinoda, Sadaoki Furui
Citation(English)	Proc. TRECVID BBC Rushes Summarization Workshop (TVS 2008) at ACM Multimedia, Vol. , No. , pp. 129-133
Pub. date	2008, 10

In Proceedings of the TRECVID BBC Rushes Summarization Workshop
(TVS 2008) at ACM Multimedia, New York, USA, pp.129–133 (2008–10)

Automatically Estimating Number of Scenes for Rushes Summarization

Koji Yamasaki, Koichi Shinoda and Sadaoki Furui
Department of Computer Science, Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku,Tokyo, Japan
yamasaki@ks.cs.titech.ac.jp, {shinoda, furui}@cs.titech.ac.jp

ABSTRACT

This paper describes our video summarization system using a model selection technique to estimate the optimal number of scenes for a summary. It uses a minimum description length as a model selection criterion and carries out two-stage estimation. First, we estimate the number of scenes in each shot, and then we estimate the number of scenes in a whole video clip. We model a set of scenes with a Gaussian mixture model, where the mixture component is assumed to represent one scene. Our system was evaluated in the TRECVID 2008 rushes summarization task, where the test video set was unedited materials provided by the BBC. Our scores were about the same as the average of all the participants for the eight evaluation measures.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing

General Terms

Algorithms, Performance

Keywords

Video summarization, TRECVID, rushes, model selection, MDL

1. INTRODUCTION

Recent advances in computer technology, particularly in storage and network technology, have resulted in significant increases in the amount and quality of video content. While users have been able to access a large amount of video data, browsing the entire content of video databases has become difficult. Therefore, content-based analysis and retrieval techniques for video content have been studied extensively [7, 13, 11].

TVS'08, October 31, 2008, Vancouver, British Columbia, Canada.

Our goal was to develop a summarization system to support video editing. Today, the spread of digital video cameras and the reduced price and advanced functionality of video editing software enable general users to edit videos. However, the efficiency of access to desired video scenes and speed of browsing by video technology are not sufficient. Therefore, we try to improve the efficiency of editing work by making a summary composed of the necessary scenes for the user to grasp the overall content.

TRECVID workshop is held every year to promote progress in content-based analysis and retrieval from digital video [14]. In 2007, a new rushes summarization task was started with the goal to make a summary of BBC rushes video. There were mainly two approaches in TRECVID 2007. The first was to eliminate junk scenes [10, 16]. Because rushes video clips used in TRECVID are unedited, they contain many junk scenes, such as colorbars, clapboards, and completely black shots. Pan *et al.* detected these scenes using a combination of video and audio features [10]. The second approach was to extract the necessary scenes for the overall content to be grasped [4, 15]. Liu *et al.* developed a system relying on speech and face detection to create a human-centric video summary [4]. Zhang *et al.* calculated an importance function for segments along the whole run of the video using a combination of face detection, camera motion detection, and an audio excitement analysis algorithm [15].

None of them, however, focused on the number of scenes to be selected for a summary. The number of necessary scenes differs in each video. A long video clip does not always need many scenes for its content to be understood, nor does a short video clip always need only a few scenes. For example, an animal documentary may be a long video, but necessary scenes might be very few. Therefore, we think estimating the optimal number of scenes is necessary for making a better summary.

In this study, we focus on estimating the number of scenes necessary to grasp the overall content. We estimated the number of scenes in the following two stages using minimum description length (MDL) [12] as a model selection criterion. In the first stage, the segments within each shot are clustered into several scenes. In the second stage, similar scenes are clustered and the optimal number of scenes for a summary is estimated.

This paper is organized as follows. Section 2 briefly describes the overall approach of the system , Section 3 explains the video features that we use, and Section 4 explains how we measure scene similarity. Then, Section 5 explains how we estimate the optimal number of scenes for a sum-

mary and Section 6 explains how we generate a summary. Section 7 reports our experimental results. Finally, Section 8 summarizes our work.

2. SYSTEM OVERVIEW

Our system consists of two stages for clustering and estimating the optimal number of scenes (Fig. 1). As preparation, a video clip is first segmented into several camera shots. Then, each shot is divided into segments with equal length K. We perform shot boundary detection by a simple method using a support vector machine (SVM) [6]. We also remove shots with durations shorter than a predetermined threshold (set to K in our system) because such short shots rarely convey important information. In the first stage, we cluster segments within a shot into several scenes by a bottom-up clustering method using Jensen-Shannon divergence [3] and estimate the optimal number of clusters (scenes) by MDL criterion. We model a shot with a Gaussian mixture model (GMM), where each component is expected to represent one scene, and estimate the optimal number of mixtures by MDL criterion. After we perform segment clustering for all shots, in the second stage we cluster the resultant scenes and estimate the optimal number of scenes for a whole video clip in a similar way. Finally, we extract a segment that contains the closest frame to the centroid from each scene to generate a summary.

We carry out clustering in two stages, segment clustering for each shot and scene clustering for the full video, for the following two reasons. The first is to reduce in the computational complexity. We use bottom-up clustering, where we need to compute the distances between all scenes. The computational cost to cluster scenes becomes much smaller in this two-stage clustering. The second is the different characteristics of the units to be clustered. In the first stage, we cluster video *segments* with equal length to make several scenes. In the second stage, we cluster *scenes* to bring similar scenes together and remove repeated scenes in the summary. The parameters used for clustering might be different between these two stages.

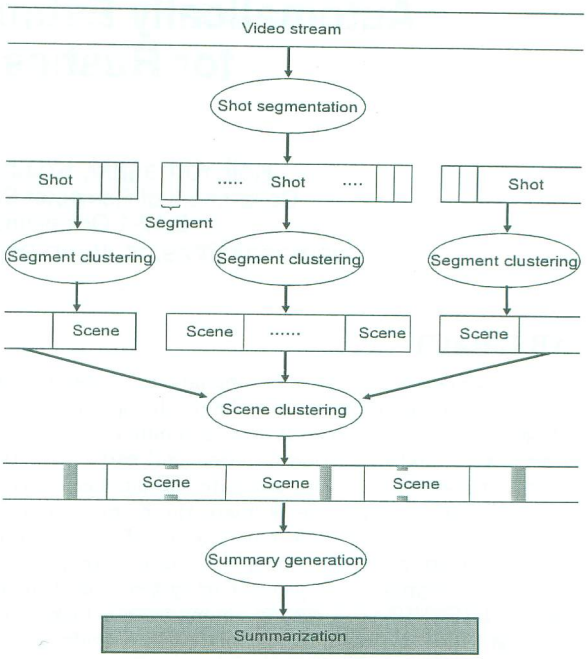

Figure 1: Our system overview.

3. FEATURES

We use a YCbCr color histogram and optical flow as video features. For each video clip, we normalize the value of each feature such that its distribution has mean 0 and variance 1. This is because the dynamic range and the mean of the feature values are generally different between video clips.

3.1 YCbCr color histogram

YCbCr is a family of color spaces used in video systems. Y is the luminance component and Cb and Cr are the blue and red chrominance components, respectively. To reduce the number of dimensions, our system computes a histogram of 8 bins for the number of pixels only in Y space. The rough constitution of each frame can be grasped through this luminance information.

3.2 Optical flow

Optical flow is the velocity field that warps one image into another. We use Lucas-Kanade point-based tracking functions [5, 2] included in the OpenCV [8]. We split each video frame into 2×2 grids. We also use a region with the same size at the center. Then, we carry out the following process for each of these five regions. First, luminance is extracted from each RGB image to create gray-scale images. Then, an eigenvalue-based method is used to identify corner locations. For each point successfully tracked, a vector corresponding to its displacement between frames is computed. Let $p_n = [p_{x_n}\ p_{y_n}]^\top$ denote each candidate point on the current frame and $p'_n = [p'_{x_n}\ p'_{y_n}]^\top (n = 1, ..., N)$ denote its corresponding point on the next frame. Optical flow vector $v_n = [v_{x_n}\ v_{y_n}]^\top$ of point n is given by

$$v_n = \begin{bmatrix} v_{x_n} \\ v_{y_n} \end{bmatrix} = \begin{bmatrix} p'_{x_n} \\ p'_{y_n} \end{bmatrix} - \begin{bmatrix} p_{x_n} \\ p_{y_n} \end{bmatrix}. \quad (1)$$

The mean $\mu = [\mu_x\ \mu_y]^\top$ of the optical flow vectors,

$$\mu = \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix} = \begin{bmatrix} \frac{\sum_{n=0}^{N} v_{x_n}}{N} \\ \frac{\sum_{n=0}^{N} v_{y_n}}{N} \end{bmatrix}, \quad (2)$$

represents a camera shift, such as pan and tilt.

4. SCENE SIMILARITY

To measure the similarity between scenes, we use Jensen-Shannon divergence [3]:

$$\mathrm{JS}(p\|q) = \frac{\mathrm{KL}(p\|q) + \mathrm{KL}(q\|p)}{2}, \quad (3)$$

where $\mathrm{KL}(p\|q)$ is a Kullback-Leibler divergence [12] between distribution p and q. We calculate Kullback-Leibler diver-

gence using a closed formed expression:

$$\mathrm{KL}(p\|q) = \frac{1}{2}\Big[\log\frac{\Sigma_q}{\Sigma_p} + \mathrm{Tr}[\Sigma_q^{-1}\Sigma_p] - d + (\mu_p-\mu_q)^T\Sigma_q^{-1}(\mu_p-\mu_q)\Big], \quad (4)$$

where μ_p and μ_q are the mean vectors for distributions p and q respectively and Σ_p and Σ_q are covariance matrices for each distribution. Kullback-Leibler divergence is often used for measuring the similarity between two probability distributions, but it is not symmetric. Jensen-Shannon divergence is its symmetrized version.

5. MODEL SELECTION FOR CLUSTERING

We cluster scenes using the scene similarity measure defined in the previous section and estimate the optimal number of scenes using MDL criterion. In the first stage, we divide each shot into segments with equal numbers of frames and cluster those segments into several scenes as follows.

Step 0 Assume each segment corresponds to one cluster.

Step 1 Calculate the distance between all the clusters.

Step 2 Choose the closest two clusters and model all segments in these two clusters with a single Gaussian.

Step 3 If $N_c > T_{\max}$, return to Step1,
else if $T_{\min} \le N_c \le T_{\max}$, go to Step 4,
else if $N_c < T_{\min}$, go to Step 5.
Here N_c is the number of clusters, $T_{\max}$ is the predetermined maximum number of clusters and $T_{\min}$ is the predetermined minimum number of clusters.

Step 4 Bring together all the Gaussian distributions obtained in the previous stage to form a GMM. Calculate description length, $\mathrm{DL}(N_c)$, as follows:

$$\mathrm{DL}(N_c) = -l + \lambda p, \quad (5)$$

$$p = \frac{1}{2}[(c-1)+c\{n+\frac{1}{2}n(n+1)\}]\log N, \quad (6)$$

where l is the log likelihood of the GMM, N is the total number of frames, c is the mixtures of the GMM, and n is the dimension number of features. p is a penalty that is an increasing function of the number of clusters (scenes), and λ is a predetermined weight for adjusting the estimated number of scenes.
Go to Step 1.

Step 5 Choose the optimal number of clusters by the following formula:

$$\hat{N_c} = \underset{N_c}{\mathrm{argmin}}\,\mathrm{DL}(N_c)\ . \quad (7)$$

We assume that the number of mixtures in the GMM that minimizes the value of DL corresponds to the optimal number of clusters in the shot.

It should be noted that the weights for each component in a GMM are not estimated using the conventional EM algorithm. An i-th mixture weight w_i is calculated as follows:

$$w_i = f_i/f_{\mathrm{shot}}, \quad (8)$$

Table 1: Results in TRECVID2008 rushes summarization.

Criterion	DU	XD	TT	VT	IN	JU	RE	TE
Baseline	33.9	0.4	58.7	34.7	0.83	2.3	2.0	1.3
Mean	28.4	3.8	40.4	30.6	0.44	3.2	3.3	2.7
Our result	32.4	1.6	41.7	34.3	0.47	2.7	3.3	3.0

where f_i is the number of frames in the i-th cluster and f_{shot} is the number of frames in the shot. This is mainly to reduce the computational costs.

In the second stage, we bring similar scenes together in a whole video clip in a similar way, except for the differences in the control parameters of λ, $T_{\max}$ and $T_{\min}$.

6. SUMMARY GENERATION

For each scene obtained in the previous section, we select a segment that contains the closest frame to the scene's centroid from its corresponding segments to generate a summary because it likely best represents the scene. Due to the constraint on the length of the summary, only f_{scene} frames in the selected segment are used.

$$f_{\mathrm{scene}} = \begin{cases} K & (K \le f_c/s_o), \\ f_c/s_o & (\text{others}), \end{cases} \quad (9)$$

where f_{scene} is the number of frames composing one scene, f_c is the maximum number of frames for a summary of a video clip, and s_o is the estimated number of scenes by MDL criterion. The maximum number of summary frames should be 2% of all the frames in the TRECVID 2008 evaluation.

7. EXPERIMENT

We evaluated our proposed system on the TRECVID 2008 test set [9]. We predetermined the parameters used in our algorithm as follows: the number of frames in a segment was $K = 50$, the maximum and minimum number of clusters in the segment clustering were $T_{\max} = 10$ and $T_{\min} = 1$, the maximum and minimum number of clusters in the scene clustering were $T_{\max} = 50$ and $T_{\min} = 5$, and the weight of penalty in Eq. (5) in both the segment and scene clustering was $\lambda_1 = \lambda_2 = 3$.

Table 1 shows our results in TRECVID 2008 compared with the baseline and mean of all participants. In most measures, our proposed system obtained scores close to the means. However, JU was significantly lower because our summaries contained many junk scenes such as color bar and clapboard scenes. This may be due to the following two reasons. First, while our system eliminated junk scenes shorter than a given threshold, other long junk scenes remained and were clustered. Second, our system was not able to exclude unnecessary scenes such as clapboards scenes well because we used only two low-level features.

Although our system's other scores were almost the same as the respective means, it is not yet confirmed that our system successfully estimated the optimal number of scenes for a summary. The difference between the estimated number of scenes and the ground truth number of scenes is shown in Fig. 2. The number of scenes was overestimated for most of the video clips. Our system used only two low-level features, the YCbCr color histogram and optical flow. These two features might not be able to capture all the characteristics of scenes. In addition, the normalization method might

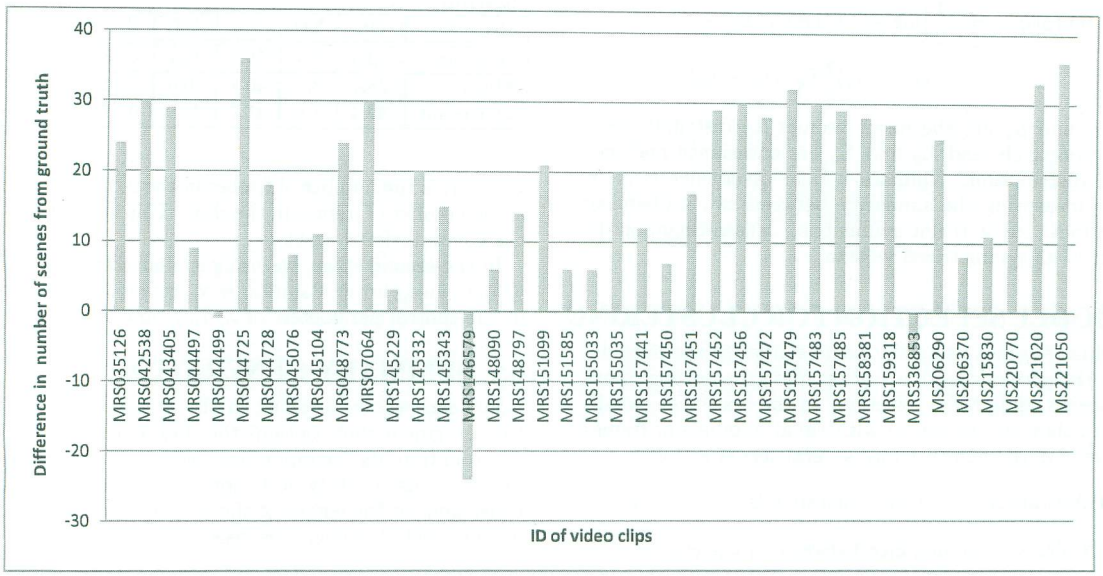

Figure 2: Difference between number of scenes estimated by our system and ground truth in TRECVID 2008 test dataset.

also be one of the causes of this over-estimation. Because we normalized features so that the distribution of each video clip had mean 0 and variance 1, the variance between scenes was very small. Therefore, in the calculation of description length, the log-likelihood of the first term in Eq. (5) became too large compared with the second penalty term.

8. CONCLUSIONS AND FUTURE WORK

We described a video summarization system using MDL criterion to estimate the optimal number of scenes for a summary. Each video segment/scene is modeled with a single Gaussian and each shot/video clip is modeled with a GMM. Then, we choose the number of mixtures that minimizes the value of the description length to be the optimal number of scenes. Our system obtained a score close to the mean of each measure except JU.

In the future, we will try to improve our system in several ways. First, we plan to adapt other techniques of removing junk scenes because our summaries still contained many and our JU score was significantly lower than the mean. Second, we plan to add other features to our system to better capture the characteristics of scenes. Finally, we plan to use other model selection criteria, such as Akaike Information Criterion (AIC) [1] and a variational Bayesian method, to make a better video summary.

9. ACKNOWLEDGEMENTS

This work was supported by Grant-in-Aid for Scientific Research (B) (20300063) and the Microsoft IJARC CORE4 project.

10. REFERENCES

[1] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, AC-19(6):716–723, 1974.

[2] J.-Y. Bouget. Pyramidal implementation of the lucas kanade feature tracker description of the algorithm. Technical report, Intel Corporation Microprocessor Research Labs, 2000.

[3] J. Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991.

[4] Z. Liu, E. Zavesky, D. Gibbon, B. Shahraray, and P. Haffner. AT&T research at TRECVID 2007. In *TREC Video Retrieval Evaluation Online Proceedings*, 2007.

[5] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *Proc. 7th International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[6] T. Nakamura, Y. Miyamura, K. Shinoda, and S. Furui. TokyoTech's TRECVID2006 notebook. In *TREC Video Retrieval Evaluation Online Proceedings*, 2006.

[7] T. Nakamura, K. Shinoda, and S. Furui. TokyoTech's TRECVID2007 notebook. In *TREC Video Retrieval Evaluation Online Proceedings*, 2007.

[8] *Intel Open Source Computer Vision Library.* http://www.intel.com/research/mrl/research/opencv/.

[9] P. Over, A. F. Smeaton, and G. Awad. The TRECVid 2008 BBC rushes summarization evaluation. In *TVS '08: Proceedings of the International Workshop on TRECVID Video Summarization*, pages 1–20, New York, NY, USA, 2008. ACM.

[10] C.-M. Pan, Y.-Y. Chuang, and W. H. Hsu. NTU

TRECVID-2007 fast rushes summarization system. In *TVS '07: Proceedings of the international workshop on TRECVID video summarization*, pages 74–78, New York, NY, USA, 2007. ACM.

[11] Z. Pan and C.-W. Ngo. Moving-object detection, association, and selection in home videos. *IEEE Transactions on Multimedia*, 9(2):268–279, February 2007.

[12] J. Rissanen. A universal prior for integers and estimation by minimum description length. *Ann. Statist.*, 11(2):416–431, 1983.

[13] K. Shinoda, K. Ishihara, S. Furui, and T. Mochizuki. Automatic score scene detection for baseball video. *International Symposium on Large-Scale Knowledge Resources (LKR2008)*, pages 226–240, March 2008.

[14] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM Press.

[15] E. Spyrou, P. Kapsalas, G. Tolias, P. Mylonas, and Y. A. et al. The COST292 experimental framework for TRECVID 2007. In *TREC Video Retrieval Evaluation Online Proceedings*, 2007.

[16] F. Wang and C.-W. Ngo. Rushes video summarization by object and event understanding. In *TVS '07: Proceedings of the international workshop on TRECVID video summarization*, pages 25–29, New York, NY, USA, 2007. ACM.