SHOT BOUNDARY DETECTION USING TEMPORAL STATISTICS MODELING

Xiaoming Liu and Tsuhan Chen

Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, U.S.A.
xiaoming@andrew.cmu.edu   tsuhan@cmu.edu

ABSTRACT

In multimedia information retrieval, shot boundary detection is a very active research topic. In order to perform shot boundary detection, we propose an algorithm for modeling temporal statistics using a novel eigenspace updating method. The feature extracted from the current frame is compared with a model trained from features in the previous frames. A shot boundary is detected if the new feature does not fit well to the existing model. The model is based on principal component analysis (PCA), or the eigenspace method, in which the eigenspace can be updated to capture the non-stationary statistics of the features. The experiment results show that the proposed algorithm outperforms the traditional direct differencing method.

1. INTRODUCTION

The increased availability and usage of digital video have created a need for automated video content analysis techniques. These include automatically detecting the boundaries between video shots. A shot in a video sequence refers to a contiguous recording of one or more video frames depicting a continuous action in time and space [1]. In a video database, the isolation of shots is of interest because the shot level organization of video sequences is considered appropriate for video browsing and content based video retrieval [2].

Many researchers have proposed algorithms to perform shot boundary detection based on certain features extracted from video frames, such as pixel differences [3], statistical differences [4], the histogram [5], compression differences [6], etc. However, most of these approaches focus on choosing the feature representation from adjacent frames to detect shots boundaries. If there is a large change between the feature of the current frame and that of the previous frame, a shot boundary is detected. We call this the *direct differencing method*. Not much prior work has taken advantage of the temporal characteristics carried by the video sequence. We propose to detect shot boundaries by comparing the difference between the current frame and a model trained from multiple previous frames. Intuitively speaking, when there is a new shot appears, the content of the new shot differs from that of the previous shot, instead of only that of the previous frame. By comparing the current frame with a model of multiple previous frames, the detection scheme can be more tolerant to the intra-variation within one shot, such as the variation caused by the camera panning or object motion. For example, Figure 1 illustrates the feature value as a function of time in a video sequence. Mostly likely the direct differencing method will incorrectly consider F_1, F_2 and F_4 as shot boundaries because there is a big change in the feature value compared to the previous frame, while the actual shot boundary, F_3, might be missed since it does not show big change compared to its previous frame. However, based on a model trained from multiple previous frames, it is likely that we would not detect F_1, F_2 or F_4, and we would detect F_3 correctly as a shot boundary because the model manifests the statistics of the previous frames.

In this paper, we propose to model temporal statistics using PCA, or the eigenspace method [7]. Once the feature, such as the histogram, has been extracted from the current frame, it will be compared with the eigenspace model, which is trained from features in the previous frames. A shot boundary is detected when the feature does not fit well to the existing model, which is measured by the difference between the current frame and the eigenspace.

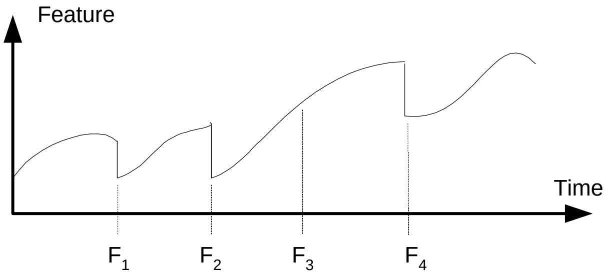

Figure 1 Illustration of shot boundary detection

The eigenspace method has been used in many fields, such as data compression [8], feature extraction [9], and object recognition [10]. Murakami and Kumar proposed the first eigenspace updating algorithm [11] to train the eigenspace on the fly as new samples come in. Chandrasekaran et al. proposed another updating algorithm based on Singular Value Decomposition (SVD) [12]. However, both algorithms assume stationary feature statistics. For the non-stationary case, the eigenspace should be based more on the statistics of recent samples and less on the statistics of older samples. In this paper, we propose a novel algorithm to accomplish this.

The paper is organized as follows. The shot boundary detection algorithm will be presented in Section 2. In Section 3, we will introduce the proposed statistics modeling method via eigenspace updating. In Section 4, we compare the detection performance between our algorithm and the traditional direct differencing method using realistic test video sequences. The conclusion and future work are presented in Section 5.

2. SHOT BOUNDARY DETECTION ALGORITHM

In the shot boundary detection algorithm, when a new frame F_n becomes available, its histogram representation X_n is computed. Suppose we already train an eigenspace based on histograms of the previous frames, $F_{n-1}, F_{n-2}, \dots$, using our modeling algorithm introduced in the next section. Now the new histogram X_n is projected into the existing eigenspace via the following equation:

$$w_i = \phi_{n-1}^{(i)T}(X_n - \hat{M}_{n-1}) \quad i = 1,2,\dots,N \tag{1}$$

where $\phi_{n-1}^{(i)}$ is the *i*-th eigenvector of the eigenspace at time $n-1$, $\hat{M}_{n-1}$ is the mean histogram at time $n-1$, N is the dimension of the eigenspace, and w_i is the *i*-th coefficient of X_n in the eigenspace. Based on the coefficients, we can reconstruct the histogram X_n by

$$\hat{X}_n = \sum_{i=1}^{N} w_i \phi_{n-1}^{(i)} + \hat{M}_{n-1} \tag{2}$$

The difference between X_n and $\hat{X}_n$, $\varepsilon_n = \left\| X_n - \hat{X}_n \right\|^2$, can then be used to measure how well the current histogram X_n can fit to the eigenspace model.

If the current frame is the start frame of a new shot, we will obtain a relatively large ε_n. Therefore, a particular frame F_n is detected as a shot boundary if ε_n is larger than a pre-defined threshold.

3. MODELING OF TEMPORAL STATISTICS

We propose two algorithms for modeling temporal statistics. When the dimension of the eigenspace is close to the dimension of the feature space, we will perform PCA based on updating a covariance matrix, which will be introduced in Section 3.1. If the dimension of the eigenspace is much smaller than the dimension of the feature space, an algorithm based on updating an inner-product matrix will be used, which is introduced in Section 3.2.

3.1 Updating the covariance matrix

Since PCA is to determine the eigenvectors given samples in the feature space, the first step in PCA is to estimate the mean and covariance of the samples. Let X_n denote the random process, where each X_n is a d-dimension feature vector. We estimate the mean, $\hat{M}_n$, of this random process at each time n as follows.

$$\hat{M}_n = \frac{X_n + \alpha_m X_{n-1} + \alpha_m^2 X_{n-2} + \cdots}{1 + \alpha_m + \alpha_m^2 + \cdots} \tag{3}$$

where α_m is called the *decay parameter*. It controls how much the previous samples contribute to the estimation of the current mean compared to the current sample. Assume 0< α_m <1. The mean estimator can be formed as:

$$\hat{M}_n = \alpha_m \hat{M}_{n-1} + (1 - \alpha_m) X_n \tag{4}$$

which shows that based on the current sample and the previous mean, we can obtain the new mean in a recursive manner. How to choose α_m mainly depends on the knowledge of the random process. If the statistics of this random process change fast, we will choose a smaller α_m. If the statistics change slowly, a larger α_m will perform better.

Similarly, the covariance matrix $\hat{C}_n$ can be estimated as follows:

$$\hat{C}_n = \alpha_v \hat{C}_{n-1} + (1 - \alpha_v)(X_n - \hat{M}_n)(X_n - \hat{M}_n)^T \tag{5}$$

Here α_v is also a decay parameter. Now we have $\hat{C}_n$ at time n, we can perform PCA for $\hat{C}_n$ and obtain the corresponding eigenvectors. We keep N eigenvectors corresponding to the N largest eigenvalues. In the recursive updating process, we only need to store the mean vector $\hat{M}_n$ and the covariance matrix $\hat{C}_n$. All the previous samples can then be thrown away.

3.2 Updating the inner-product matrix

If the dimension of the feature space d is large, it is very inefficient to store and update the covariance matrix $\hat{C}_n$.

To solve this problem, we propose a modeling algorithm based on updating the inner-product matrix.

Suppose at time n, we already have done PCA for the random process till time $n-1$. Thus we have eigenvectors $\phi_{n-1}^{(i)}$ and eigenvalues $\lambda_{n-1}^{(i)}$ of the covariance matrix $\hat{C}_{n-1}$, which can be expressed as

$$\hat{C}_{n-1} = \lambda_{n-1}^{(1)}\phi_{n-1}^{(1)}\phi_{n-1}^{(1)^T} + \lambda_{n-1}^{(2)}\phi_{n-1}^{(2)}\phi_{n-1}^{(2)^T} + \cdots + \lambda_{n-1}^{(d)}\phi_{n-1}^{(d)}\phi_{n-1}^{(d)^T} \qquad (6)$$

where the eigenvalues, $\lambda_{n-1}^{(i)}$, have been sorted in decreasing order and the superscript indicates the order of eigenvalues. By retaining only the first Q eigenvectors (with the largest eigenvalues), we can approximate $\hat{C}_{n-1}$ as

$$\hat{C}_{n-1} \approx \lambda_{n-1}^{(1)}\phi_{n-1}^{(1)}\phi_{n-1}^{(1)^T} + \lambda_{n-1}^{(2)}\phi_{n-1}^{(2)}\phi_{n-1}^{(2)^T} + \cdots + \lambda_{n-1}^{(Q)}\phi_{n-1}^{(Q)}\phi_{n-1}^{(Q)^T} \qquad (7)$$

The criteria for choosing Q vary, and depend on practical applications. Now we can use (4) to estimate the mean at time n. By replacing $\hat{C}_{n-1}$ in (5) with (7), we can obtain

$$\hat{C}_n = B_n B_n^{\ T} \qquad (8)$$

where

$$B_n = \left[\sqrt{\alpha_v \lambda_{n-1}^{(1)}}\phi_{n-1}^{(1)} \quad \cdots \quad \sqrt{\alpha_v \lambda_{n-1}^{(N)}}\phi_{n-1}^{(N)} \quad \sqrt{1-a_v}(X_n - \hat{M}_n) \right] \qquad (9)$$

An inner-product matrix can be formulated as

$$A_n = B_n^T B_n \qquad (10)$$

Elements of A_n can be described as follows:

$$\begin{aligned} (A_n)_{i,j} &= \alpha_v \sqrt{\lambda_{n-1}^{(i)}\lambda_{n-1}^{(j)}}\delta_{ij}; \quad i,j = 1,2,...,Q \\ (A_n)_{i,Q+1} = (A_n)_{Q+1,i} &= \sqrt{\alpha_v(1-\alpha_v)\lambda_{n-1}^{(i)}}(X_n - \hat{M}_n); \quad i = 1,2,...,Q \\ (A_n)_{Q+1,Q+1} &= (1-\alpha_v)(X_n - \hat{M}_n)^T(X_n - \hat{M}_n). \end{aligned} \qquad (11)$$

Since A_n is a matrix with the size of $Q+1$ by $Q+1$, smaller than $\hat{C}_n$, we can easily determine its eigenvectors ψ_n, which satisfy

$$B_n^T B_n \psi_n^{(i)} = \lambda_n^{(i)}\psi_n^{(i)} \quad i = 1,2,...,Q+1 \qquad (12)$$

Pre-multiplying (12) with B_n, we obtain the eigenvectors of $\hat{C}_n$ as follows:

$$\phi_n^{(i)} = \lambda_n^{(i)^{-\frac{1}{2}}} B_n \psi_n^{(i)} \quad i = 1,2,...,Q+1 \qquad (13)$$

We summarize the iterative modeling algorithm as follows:

***Initialization**:*

1. Given the first two samples X_0, X_1, estimate the mean, $\hat{M}_1$, using (4), and construct the matrix

$$B_1 = \left[\sqrt{\alpha_v}(X_0 - \hat{M}_1) \quad (X_1 - \hat{M}_1) \right] \qquad (14)$$

2. Based on (12) and (13), we can get the eigenvector ϕ_1, and the eigenvalue λ_1.

***Iterative updating**:*

1. Get a new sample X_n.
2. Estimate the mean, $\hat{M}_n$, at time n by (4), and get B_n from (9).
3. Form the matrix A_n as in (11) and calculate its eigenvectors ψ_n and eigenvalues λ_n.
4. Sort the eigenvalues λ_n, and retain Q eigenvectors corresponding to the largest eigenvalues.
5. Obtain the eigenvectors at time n using (13).

Due to the approximation in (7), among the Q eigenvectors, typically the first few eigenvectors are more precise than the others. Therefore, in practice if we want to use N eigenvectors for shot boundary detection, we would keep Q to be a number larger than N.

4. EXPERIMENTS

In this section, we compare the shot boundary detection performance between our algorithm and the traditional direct differencing algorithm. As a statistics modeling method, our algorithm can be applied on any features that are suitable for shot boundary detection. In other words, the eigenspace can be used to model any features known in literature, such as the histogram, DCT coefficients [13], etc. In the following experiments we choose the histogram as the feature to work on.

We collect four video sequences as the testing data, which come from news videos and video recording of meetings. The snapshots of the testing videos are shown in Figure 2. Each frame is represented as a histogram with 256 bins, which is modeled using the algorithm introduced in Section 3.2. The performance is expressed in terms of recall and precision of shot boundary detection. By tuning a threshold, we can generate a recall-precision curve for each testing sequence. Given multiple recall-precision curves, we can average the precision values at each specific recall value, to obtain the recall-precision curve of the whole system. The experiment results are shown in Figure 3. Better detection performance is observed from the whole range of recall values. Especially when the recall value is closer to 1 or when the precision is maximized, which are the operation points that most of the systems prefer to work at, our algorithm has about 5~13% better performance than the direct differencing method.

Figure 2 Snapshots of testing video sequences

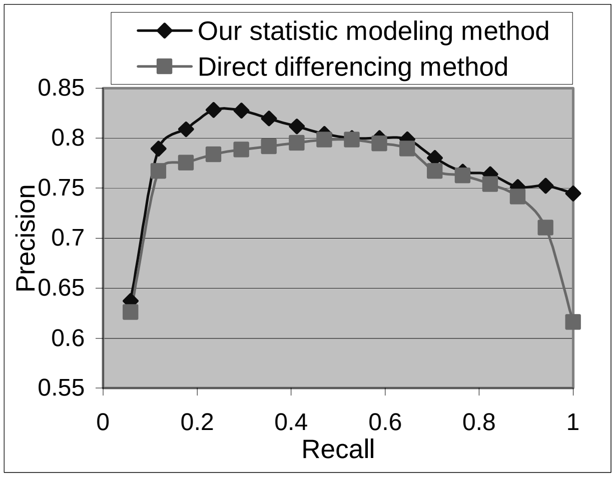

Figure 3 Experiment results of shot boundary detection

5. CONCLUSION

We proposed to perform shot boundary detection by modeling temporal statistics via eigenspace updating. The experiment results showed superior performance of the proposed algorithm compared to the traditional direct differencing method.

There are several ways to extend our work. The first way is that we can apply our algorithm in the reverse order of the video sequence, i.e., using the current feature to compare with the model trained from future frames. Combining the information from the modeling of two directions will improve the detection. The other way is that for each frame, two statistics models can be built: one for the previous frames, the other for the current and future frames. Thus the difference between the parameters of two models can be a useful cue for shot boundary detection.

Although we only show our algorithm in shot boundary detection based on the histogram, our approach can be applied to other features, such as DCT coefficients. It can be used in other applications as well, such as the detection of facial expression changes.

6. REFERENCES

[1] D. Arijon, Grammar of Film Language. Los Angles: Silman_James Press, 1976.

[2] S. Smoliar and H. Zhang, “Content-based video indexing and retrieval”, IEEE Multimedia, Vol. 1, pp. 62-72, 1994.

[3] H. Zhang, A. Kankanhalli and S. Smoliar, “Automatic Partitioning of Full-motion Video”, Multimedia Systems, Vol.1, No.1, pp.10-28, 1993.

[4] R. Kasturi and R. Jain, “Dynamic Vision”, in Computer vision: Principles, R. Kasturi and R. Jain, Editors, IEEE Computer Society Press, Washington, 1991.

[5] A. Nagasaka and Y. Tanaka, “Automatic Video Indexing and Full-Video Search for Object Apprearances”, in Visual Database System II, E. Knuth, L. Wegner, Editors, Elsevier Science Publishers, pp.113-127, 1992.

[6] T. Little, G. Ahanger, R. Folz, J. Gibbon, F. Reeve, D. Schelleng and D. Venkatesh, “A Digital On-Demand Video Service Supporting Content-Based Queries”, Proc. ACM Multimedia 93, Anaheim, CA, pp.427-436, August, 1993.

[7] Y.T.Chien, K.S. Fu, “On the generalized Karhunen-Loeve expansion”, IEEE Transaction on Information Theory, Vol.13, No.3, pp.518-520, July, 1967.

[8] A. Habibi P.A. Wintz, “Image coding by linear transformation and block quantization techniques”, IEEE Transaction on Communication and Technology. Vol. CoOM-19, pp.948-956, 1971.

[9] R. J. Wong, P. A. Wintz, “Information extraction, SNR improvement, and data compression in multispectral imagery”, IEEE Transaction on Communication and Technology. Vol. CoOM-21, pp.1123-1131, 1973.

[10] E. Oja, Subspace methods of pattern recognition. Letchworth, Hertfordshire, England. New York: Wiley, 1983.

[11] H. Murakami, B.V.K.V. Kumar, “Efficient calculation of primary images from a set of images”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.4, No.5,pp.511-515, Sept. 1982.

[12] S. Chandrasekaran, B.S. Manjunath, Y.F. Wang, J. Winkeler, H. Zhang, "An eigenspace update algorithm for image analysis", Graphical Models and Image Processing, Vol.59, No.5, Academic Press, pp.321-332, Sept. 1997.

[13] C. Chang and S. Lee, “Video content representation, indexing, and matching in video information systems”, Journal visual communication and image representation, Vol.8, No.2, pp.107-120, June, 1997.