A Robust Dissolve Detector by Support Vector Machine

Chong-Wah Ngo
Department of Computer Science
City University of Hong Kong
cwngo@cs.cityu.edu.hk

ABSTRACT

In this paper, we propose a novel approach for the robust detection and classification of dissolve sequences in videos. Our approach is based on the multi-resolution representation of temporal slices extracted from $3D$ image volume. At the low-resolution (LR) scale, the problem of dissolve detection is reduced as cut transition detection. At the high-resolution (HR) space, Gabor wavelet features are computed for regions that surround the cuts located at LR scale. The computed features are then input to support vector machines for pattern classification. Encouraging results have been obtained through experiments.

Categories and Subject Descriptors

I.2.10 [**Vision and Scene Understanding**]: Video Analysis

General Terms

Algorithms, Design, Reliability, Experimentation

Keywords

Dissolve Detector, Temporal Slices, Support Vector Machine

1. INTRODUCTION

Due to the advance of video production technology, various type of video edits (*e.g.,* cut, wipe and dissolve) can be easily created to indicate the change of space and time. By detecting these edits (shot boundaries), we can facilitate the content analysis, indexing, browsing and retrieval of video data. To date, numerous approaches have been proposed for the detection and classification of shot boundary transitions. While *cuts* can be successfully located by color [14, 16], edge [15], motion [2] and statistical [5, 13] analysis, the correct detection of gradual transitions (*dissolves* and *wipes*) is still remained as a difficult problem. This is not surprised since cuts can be easily identified by comparing two adjacent frames, gradual transitions, however, require the investigation of frames along a large temporal scale. Surveys and studies in [4, 11, 17] have indicated that cut detectors, in general, are reliable in most circumstances. Gradual transition detectors, on the other hand, can only handle simple examples. Most dissolve detectors can achieve either high recall or high precision, but not both [17]. In this paper, we focus our attention on the robust detection and classification of dissolve and non-dissolve patterns.

MM'03, November 2–8, 2003, Berkeley, California, USA.

2. RELATED WORKS

Most existing works [1, 6, 7, 9, 14] on the detection of dissolve transitions are based on the following model:

$$f(x,y,t) = (1-\alpha(x,y,t))g(x,y,t) + \alpha(x,y,t)h(x,y,t) \quad (1)$$

where f is a dissolved frame superimposed by two frames g and h at time t. Typically, g and h are frames from two different shots. The transition function α characterizes, either linearly or non-linearly, how f is dissolved over time as a result of mixing g and h. Usually $0 < \alpha(x,y,t) < 1$ with the condition $\alpha(x,y,t) \leq \alpha(x,y,t+1)$. Since Eqn(1) is irreversible, apparently, detecting and classifying dissolves is a difficult task. To simplify the problem of detection, various assumptions have made on Eqn(1). These assumptions have led to *plateau effect* [14] and *parabolic curve of variance* [1].

Take $\alpha(x,y,t) = \alpha(t)$, $g(x,y,t) = g(x,y)$ and $h(x,y,t) = h(x,y)$, we have

$$f(x,y,t) = (1-\alpha(t))g(x,y) + \alpha(t)h(x,y) \quad (2)$$

In other words, f is a dissolved sequence of two static shots g and h in $t = [t_1, t_2]$. Let $\mathcal{F}(t) = f(x,y,t)$, by taking the frame difference, we have

$$\frac{\mathcal{F}(t) - \mathcal{F}(t+k)}{\mathcal{F}(t-k) - \mathcal{F}(t)} = \beta(t,k) \quad (3)$$

where $\beta(t,k) = \frac{\alpha(t+k)-\alpha(t)}{\alpha(t)-\alpha(t-k)} > 1$ and $t = [t_1 + k, t_2 - k]$. If $k > t_2 - t_1 + 1$, plateau effect will be exhibited and this effect can be exploited effectively for dissolve detection [6, 14].

Eqn(2) can be further simplified by assuming $\alpha(t)$ as a linear function, $\alpha(t) = \frac{t-t_1}{t_2-t_1}$ for instance. This leads to a formula in term of variance:

$$\sigma_f(t) = (\sigma_g + \sigma_h)\alpha^2(t) - 2\sigma_g\alpha(t) + \sigma_g \quad (4)$$

where $\sigma_f(t)$, σ_g and σ_h are the variances of $f(x,y,t)$, $g(x,y)$ and $h(x,y)$. Since $\sigma_f(t)$ is a concave upward parabolic curve, dissolves can be detected simply by locating parabolic curves [1, 9]. The limitations of Eqn(3) and Eqn(4) are mainly due to the linearity assumption of $\alpha(t)$ and the static sequence assumption of shots g and h. As a result, most detectors are generally very sensitive to noise, camera and object motions.

Other interesting approaches include edge change ratio [15] and twin comparison [16]. Recently, modern machine learning (ML) approach is also adopted for dissolve detection [12]. The problem of detection is considered as a 2-class pattern classification problem. Low-level color and contrast features are extracted from image sequence for neural network learning. In this paper, we propose a ML approach based on support vector machines (SVM) for dissolve classification and detection. The novelties of our approach lie on three different aspects: i) reduce the problem of dissolve detection as cut detection in multi-resolution (MR) representation, ii) utilize Gabor wavelet features extracted from $2D$ temporal slices for pattern description, iii) effective filtering and selection of potential dissolve regions for feature extraction and pattern classification. Compared with [12], our approach is robust since i) actual dissolves will not be easily missed during the pre-filtering stage in our MR representation, ii) Gabor wavelet features which encode motion texture across different scales and rotations are more reliable to filter false matches due to camera and object motion.

3. PATTERN CLASSIFICATION

In this section, we describe our approach in encoding the dissolve patterns observed in temporal slices for classification. All the computation is done directly on the DC image volume of MPEG videos. This offers two advantages: computational efficiency since the $3D$ image volume is reduced by 64 times, and the volume is inherently smoothly.

3.1 Dissolve Patterns in Temporal Slices

Temporal slices are a set of $2D$ images in an image volume with one dimension in time t, and the other in space x or y, for instance. Typically, video shots appear as spatially uniform color-texture regions in temporal slices [9]. Each region is considered to exhibit a unique rhythm, and the change of rhythm can indicate the presence of shot boundaries. As a result, the analysis of temporal slices is an effective way of detecting shot transitions. While cuts and wipes can be detected by measuring the change of color-texture properties through image segmentation [9], dissolves, nevertheless, can not be easily located. This is because the rhythm of two adjacent shots are interwined during a dissolve where the change in coherency cannot be easily distinguished by color-texture properties.

Despite the difficulties in detecting dissolves by image segmentation, human eyes can still visually locate dissolves by observing the gradual change of coherency in temporal slices. Figure 1 shows several examples of dissolve patterns in temporal slices. The length of these dissolves, d_1 to d_6, varies from 30 to 80 frames. Basically human eyes are not only able to distinguish dissolve from non-dissolve patterns, but can also quickly identify the rough boundary of these dissolve regions. In Figure 1(a), a non-dissolve pattern is shown. The visual rhythm in this slice is basically generated by camera panning and object motion. The marked region, *non-diss*, is a false alarm that exhibits the similar statistical behavior as describe in Eqn(3) and Eqn(4). It can be easily detected as a dissolve by the algorithms based on these equations.

A careful observation on the dissolves in temporal slices can reveal the fact that dissolves usually have a specific temporal texture pattern that can help human perform pattern classification. The problem of dissolve *vs* non-dissolve pattern discrimination is similar to the text *vs* non-text or the face *vs* non-face classification. In this paper, we choose Gabor wavelet features to describe dissolve patterns for its biological relevance and technical properties. First, the Gabor wavelets are of similar shape as the receptive fields of simples cells in the primary visual cortex. Second, they are localized in both the space and frequency domains and have the shape of plane waves restricted by the Gaussian envelop functions. In fact, we have compared the performance of Gabor wavelet features with tensor histograms [10] and co-occurrence matrices by Support Vector Machines (SVM) and Neural Network (NN). Gabor wavelet features give the best performance for testing data and is the only features that can achieve 100% classification accuracy for training data in SVM and NN.

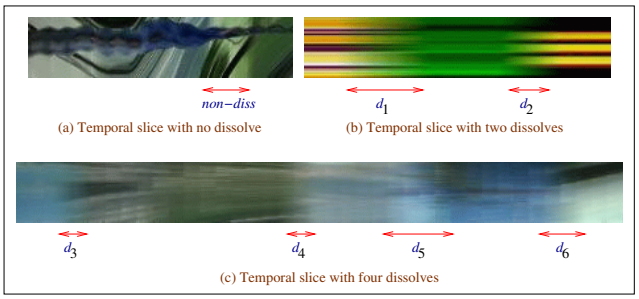

Figure 1: Dissolve and non-dissolve patterns.

3.2 Volume Processing

The length of dissolves can vary typically from 15 to 150 frames. Intuitively, a brute force approach is required in order to detect dissolves of different length. In other words, for a given image sequence, we can repeatedly compute visual features in a support window W of different temporal length, where $|W| = 15, 16, \ldots, 150$ (let $|W|$ represents the temporal length). Then, a correct dissolve should be the one with its length $n = |W|$. Nevertheless, this approach is computationally intensive. Our major observation is that, for a dissolve of length n, the dissolve is likely to be classified as a non-dissolve pattern if $|W| > n$. However, if $|W| \leq n$, the dissolve is always classified correctly. The result is not surprised since the dissolve and non-dissolve patterns are mixed when $|W| > n$.

In our approach, we set $|W| = 15$. For an $3D$ image volume of size $M \times N \times T$, a support window of $M \times N \times 15$ is slided temporally along the volume with a step size $\Delta = 3$. In each window support volume, the Gabor wavelet features are computed for the temporal slices that are extracted horizontally and vertically from the volume. The computed features of each slice are then combined as described in Section 3.3 to form a feature vector. For a dissolve with length $n > |w|$, there are $\lceil \frac{n-|W|}{\Delta} \rceil + 1$ feature vectors, each one represents a segment in the dissolve.

3.3 Gabor Wavelet Feature Extraction

Gabor wavelet feature is frequently used for browsing and retrieval of texture images, and have been shown to give good results [8]. A Gabor filter $g(x,t)$ can be written as

$$G(x,t) = (\frac{1}{2\pi\sigma_x\sigma_t}) \exp\{-\frac{1}{2}(\frac{x^2}{\sigma_x^2} + \frac{t^2}{\sigma_t^2})\} \exp\{2\pi jWx\} \quad (5)$$

where σ_x and σ_t are smoothing parameters, $j = \sqrt{-1}$, $W = \sqrt{u^2 + v^2}$ and (u, v) is the center of the desired frequency. A

self-similar filter $G_{\theta\mathcal{S}}(x,t)$ can be obtained by the appropriate rotation θ and scaling $\mathcal{S}$ of $G(x,t)$. The Gabor filtered image of a slice $\mathbf{H}$ is

$$\hat{\mathbf{H}}_{\theta\mathcal{S}} = \mathbf{H} * G_{\theta\mathcal{S}} \tag{6}$$

where $*$ is a convolution operator. A feature vector is constructed by using the mean $\mu_{\theta\mathcal{S}}$ and the standard deviation $\sigma_{\theta\mathcal{S}}$ of all $\hat{\mathbf{H}}_{\theta\mathcal{S}}$ as components. In the experiment, $\theta = 6$ and $\mathcal{S} = 2$. The resulting feature vector has length $6 \times 2 \times 2 \times 2 = 48$ in the following form

$$\underbrace{[\mu_{00}, \sigma_{00}, \mu_{01}, \sigma_{01}, \ldots \mu_{51}, \sigma_{51},}_{\text{for horizontal slices}} \quad \underbrace{\mu_{00}, \sigma_{00}, \mu_{01}, \sigma_{01}, \ldots \mu_{51}, \sigma_{51}]}_{\text{for vertical slices}}$$

Let $\mathcal{V}_i$ represents the i^{th} component of a Gabor wavelet feature vector, where $i = \{1, 2, \ldots, n\}$. Each vector is composed of $n = 48$ components. Because the range of different feature components can vary considerably, the feature component is normalized as follows

$$\mathcal{V}_i = \frac{\mathcal{V}_i - \mu_i}{\alpha_i} \tag{7}$$

where μ_i and α_i are, respectively, the mean and standard deviation of the i^{th} feature component over the training data.

3.4 Support Vector Machine

Support Vector Machines (SVM) algorithm is based on the idea of structural risk minimization. Its generation error is bounded by the sum of training error and the VC-dimension of a classifier. By minimizing the upper bound, SVM can achieve a higher generalization performance. In our application, we employ C-Support Vector Classification (C-SVC) [3] for dissolve recognition. We use RBF (radial basis function) as the kernel function to map training vectors into high dimensional feature space for classification.

4. MULTI-RESOLUTION APPROACH

Sliding a local window and computing the Gabor wavelet features in each window support volume is a computationally intensive task. To speed up the processing time, indeed we only need to compute features for regions that consist of potential dissolve patterns. In this section, we propose a novel multi-resolution approach to detect the potential dissolve regions in temporal slices.

Figure 2 illustrates the evolving of eight dissolves to camera cuts in a pyramid representation. This is carried out by reducing the temporal resolution of a slice. In this figure, the temporal slices are down-sampled respectively by 3, 7, and 15 time units while Gaussian smoothing is imposed to preserve the temporal rhythm. As observed in the figure, when the eight different dissolves arrives at different multi-resolution levels, they gradually become camera cuts depending on their temporal length. In this example, all dissolves become cuts at the top of the pyramid.

Our strategy is to detect camera cuts at the low resolution space. After detecting the transitions, the cut boundaries are projected back to the original scale. Intuitively, the projected regions contains the potential dissolve boundaries. We temporally expand the projected regions and then compute Gabor wavelet features of the regions through a support window as described in sections 3.2 and 3.3. We adopt the algorithm in [9] to detect camera cuts. The algorithm is based on a spatio-temporal slice model that utilizes the texture and color information for image segmentation. The algorithm is efficient since only three slices: horizontal, vertical and left-diagonal slices extracted from the center of an image volume, are analyzed.

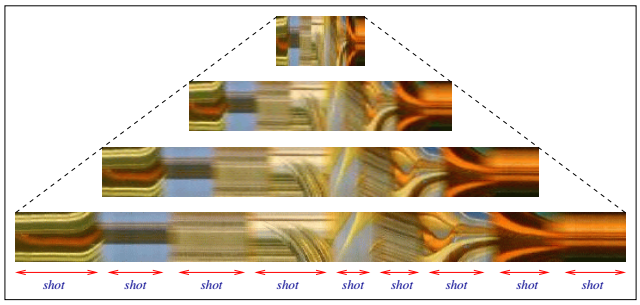

Figure 2: Evolving of *dissolves* to *cuts* in bottom-up manner along multiple scales in pyramid representation.

In our experiment, we only use two levels of pyramid. The temporal scale is down-sampled by 15 time units. At this scale, almost all dissolves are emerged as cuts in temporal slices. However, regions with fast camera and object motion are also appeared as camera cuts. By employing the algorithm in [9], we can correctly detect approximately 98% of dissolves, but only with about 20% to 30% of precision. Our goal is to filter false matches while retaining the correct dissolves through SVM classifier.

5. EXPERIMENT

The experiment is set up as follows. First we employ the cut detector in [9] to temporally partition a video into segments according to cut transitions. The slices of each segment is then temporally down-scaled by 15 time units and the same cut detector is applied to detect the cuts in the low-resolution space. The detected cuts at time t of low-resolution space are then projected to the original scale at $[15t - (7+k), 15t + (7+k)]$. The value k is a constant used to expand the projected regions. Since the exact boundary of a dissolve is always vague, adding $2 \times k$ to a projected region can increase the robustness of detection. In practice, k can always improve the recall of dissolve detection. In the experiment, we set $k = 7$. Gabor wavelet features are then computed for these projected regions and SVM classifier is employed for pattern classification.

In the experiment, eight videos are used for training while five videos are used for testing. Approximately 500 dissolves and 500 non-dissolves are used for SVM training. All dissolves are manually labeled by human subjects. The non-dissolves are basically the false alarms generated when we apply the cut detector in [9] to the eight training videos in the low resolution space. At the training stage, two-fold cross-validation and automatic parameters selection strategies are employed. About 320 patterns are selected as support vectors after the training.

We use the recall-precision measure for performance evaluation. Recall measures the capability in detecting correct dissolves, while precision measures the ability in preventing false alarms. The values of recall and precision are in the range $[0, 1]$. The values of recall and precision are combined as follows to measure the overall performance

$$\text{RP} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \tag{8}$$

Table 1: Experimental Results

Video	Frames	Dissolves	Correct Detection	False Alarm	Missed Detection	Precision	Recall	RP
1	52,561	117	101	30	16	0.77	0.86	0.81
2	24,720	135	106	18	29	0.85	0.79	0.82
3	12,588	24	19	8	5	0.70	0.79	0.74
4	25,694	77	60	18	17	0.77	0.78	0.77
5	11,571	33	28	10	5	0.74	0.85	0.79

The value of RP is high only when both recall and precision are high. Table 1 shows the experimental results of the five testing videos. In the experiment, a detection is considered correct as long as it is overlapped with a ground-truth dissolve [17]. The weighted average performance of our proposed approach on the five testing videos are: recall= 0.83, precision=0.79 and RP=0.81.

We compare our approach with an algorithm based on the variance curve [1] as stated in Eqn(4). The algorithm is implemented as follow: First, all potential dissolves are detected as described in Section 4 based on the multi-resolution approach. Then, the algorithm checks whether the potential dissolves have upward parabolic variance curves as described in Eqn(4). A dissolve is detected if there is an associated upward parabolic curve. Table 2 compares the performance of our proposed approach with the "variance curve" approach. As shown in the table, our approach is constantly better than "variance curve" in term of RP value. The main reason that "variance curve" has low recall in video-1 is that the assumptions made in Eqn(4) are violated when there are camera and object motions involved during the dissolve period. The precision is low in general since upward parabolic curves are not unique only to dissolves, they can be generated by sequences with certain camera and object motions. One simple example is a sequence with static motion, zoom, and then static again. Our proposed approach, in contrast to "variance curve", achieves satisfactory performance for both recall and precision. A close inspection of the videos reveals that false detections are mostly due to the gradual change of illumination and the gradual $3D$ rotation of objects. Missed detections are mostly due to complex scenes (*e.g.,* the dissolve region marked by $miss_2$ in Fig 3 involves a shot with flaming scene), complicated transfer functions $\alpha(x, y, t)$ in Eqn(1) (*e.g.,* the region marked $miss_1$ in Fig 3), and low contrast of two connecting shots (*e.g.,* $miss_3$ in Fig 3).

Table 2: Performance Comparison

	Proposed approach			Variance curve		
Video	Precision	Recall	RP	Precision	Recall	RP
1	0.77	0.86	0.81	0.45	0.66	0.53
2	0.85	0.79	0.82	0.82	0.78	0.80
3	0.70	0.79	0.74	0.40	0.83	0.54
4	0.77	0.78	0.77	0.48	0.81	0.60
5	0.74	0.85	0.79	0.37	0.70	0.48

6. CONCLUSION

We have presented a new approach for dissolve detection. The novelties of our approach include: dissolve pattern description by Gabor wavelet features extracted from temporal slices, potential dissolves selection by cut detection in low-resolution space, SVM based dissolve classifier. Experimental results indicate that our approach can compromise recall and precision. We believe better performance can be achieved if more training samples are included for pattern learning.

Figure 3: Missed dissolve patterns.

Acknowledgments

The work described in this paper was fully supported by RGC Grant CityU 1072/02E (Project No. 9040693).

7. REFERENCES

[1] A. M. Alattar, "Detecting and Compressing Dissolve Regions in Video Sequences with a DVI Multimedia Image Compression Algorithm", *Int. Symposium on Circuits and Systems*, vol. 1, pp. 13-16, 1993.

[2] P. Bouthemy, M. Gelgon & F. Ganansia, "A Unified Approach to Shot Change Detection and Camera Motion Characterizatio", *IEEE Trans. CSVT*, 9(7):1030-1044, 1999.

[3] C. Cortes & V. Vapnik, "Support-vector Network", *Machine Learning*, 20:273-297, 1995.

[4] U. Gargi, R. Kasturi & S. H. Straye, "Performance Characterization of Video-Shot-Change Detection Method", *IEEE Trans. CSVT.*, 10(1):1-13, Feb 2000.

[5] A. Hanjalic, "Shot Boundary Detection: Unraveled and Resolved", *IEEE Trans. CSVT*, 12(2):90-105, Feb 2002.

[6] R. A. Joyce & B. Liu, "Temporal Segmentation of Video using Frame and Histogram Space", *ICIP*, 2000.

[7] H. B. Lu, Y. J. Zhang & Y. R. Yao, "Robust Gradual Scene Change Detection", *Int. Conf. on Image Processing*, 1999.

[8] B. S. Manjunath & W.Y. Ma, "Texture Features for Browsing and Retrieval of Image Data", *IEEE Trans. on PAMI*, 18(8):837-842, Aug 1996.

[9] C. W. Ngo *et. al.*, "Video Partitioning by Temporal Slice Coherency", *IEEE Trans. CSVT*, 11(8):941-953, Aug, 2001.

[10] C. W. Ngo *et. al.*, "Motion Analysis and Segmentation through Spatio-Temporal SLices Processing", *IEEE Trans. on Image Processing*, 12(3):341-355, March 2003.

[11] R. Lienhart, "Comparison of Automatic Shot Boundary Detection Algorithms", *SPIE Proc. Storage and Retrieval for Still Image and Video Databases VII*, Jan 1999.

[12] Rainer Lienhart, "Reliable Dissolve Detection", *SPIE Proc. Storage and Retrieval for Media Database*, Jan 2001.

[13] N. Vasconcelos & A. Lippman, "Statistical Models of Video Structure for Content Analysis and Characterization", *IEEE Trans. on Image Processing*, 9(1):3-19, Jan 2000.

[14] B. L. Yeo & B. Liu, "Rapid Scene Analysis on compressed video", *IEEE Trans. CSVT*, 5(6):533-544, Dec 1995.

[15] R. Zabih *et. al.*, "A Feature-Based Algorithm for Detecting and Classifying Scene Break", *ACM Multimedia*, 1995.

[16] H. J. Zhang *et. al.*, "Automatic Partitioning of Full-motion Video", *ACM Multimedia Syst.*, 1(1):10-28, 1993.

[17] TREC-Video, http://www-nlpir.nist.gov/projects/trecvid/.