RC 21905 (98531) 5 December 2000 Computer Science

IBM Research Report

Improved Fade and Dissolve Detection for Reliable Video Segmentation

Ba Tu Truong*, Chitra Dorai, Svetha Venkatesh*
IBM Research Division
Thomas J. Watson Research Center
P. O. Box 704
Yorktown Heights, NY 10598

* Department of Computer Science
Curtin University of Technology
GPO Box U 1987 Perth 6001, Australia

Research Division
Almaden - Austin - Beijing - Haifa - T. J. Watson - Tokyo - Zurich

Improved Fade and Dissolve Detection for Reliable Video Segmentation *

Ba Tu Truong[†], Chitra Dorai[‡], Svetha Venkatesh[†]

Department of Computer Science[†]
Curtin University of Technology
GPO BOX U 1987 Perth 6001, Australia
{truongbt, svetha}@cs.curtin.edu.au

IBM T.J. Watson Research Center[‡]
P.O. Box 704, Yorktown Heights
New York 10598, USA
dorai@watson.ibm.com

Abstract

We present improved algorithms for automatic fade and dissolve detection in digital video analysis. We devise new two-step algorithms for fade and dissolve detection and introduce a method for eliminating false positives from a list of detected candidate transitions. In our detailed study of these gradual shot transitions, our objective has been to accurately classify the type of transitions (fade-in, fade-out, and dissolve) and to precisely locate the boundary of the transitions. This distinguishes our work from early work in scene change detection which focuses on identifying the existence of a transition rather than its precise temporal extent. We evaluate our algorithms against two other commonly used methods on a comprehensive data set, and demonstrate the improved performance due to our enhancements.

Key words: Video, digital television, content-based search and annotation, video segmentation, shot detection, gradual shot transition, fades, dissolves.

*Appeared in the Proceedings of the International Conference on Image Processing, Vancouver, Canada, September 2000.

1 Introduction

Video segmentation, often performed by detecting transitions occurring between shots, is a fundamental process in automatic video analysis. A shot in a video is defined as an unbroken sequence of images of a real or animated world captured between a camera's "record" and "stop" operations [1]. Shots are joined together in the editing stage of video (post) production with either sharp cuts between them or using gradual visual effects such as fades and dissolves in order to form a complete story sequence and to convey a specific meaning or a mood of the events portrayed in the video. Detecting shots and the type of transitions present between them is extremely useful in analyzing the inter- and intra-shot relationships for high level video interpretation. Locating shots aids in improving video compression [2].

The purpose of this paper is to investigate the problem of automatic detection of gradual shot transitions such as fades and dissolves. While many excellent techniques have been proposed in the literature for shot detection [2, 3, 1, 4], we found in our study of several of these that many enhancements could be incorporated to render them more accurate and robust especially when dealing with a wide range of manifestation of these effects in videos of many different genres. In detecting these gradual transitions, our primary objective is to accurately classify and measure the temporal extent of a transition. Further, we also concentrate on testing the algorithms on a comprehensive data set.

This paper proposes the following enhancements to gradual transition detection algorithms: We improve the previous work on production model based techniques [2, 3] with new two-step algorithms for detecting fades and dissolves. Instead of selecting thresholds based on the traditional *trial-and-error* approach, robust adaptive thresholds are derived analytically from the mathematical models of transitions in our approach. In addition, we also propose a simple, yet effective technique for eliminating false positives from a list of detected transitions. This verification process is performed after the fade and dissolve detectors have been executed. Finally, our algorithms have been tested on variety of data and their performance shown to be more accurate and reliable when compared with two other commonly used methods including a commercial software.

2 Improved Fade Detection Technique

During a fade, an image or a frame in a video sequence gradually darkens and is replaced by another image which either fades in or begins abruptly [5]. A fade-out occurs when the picture information gradually disappears, leaving a blank screen. A fade-in occurs when the picture gradually appears from a black screen. Enhancements to ordinary fades would include fading in/out to a white screen or other dominant colors. Fades are used often to denote time transitions. Fade-out/in combinations can also indicate relative mood and pace between shots. The fades can be used to separate different TV program elements such as the main show material from commercial blocks [6].

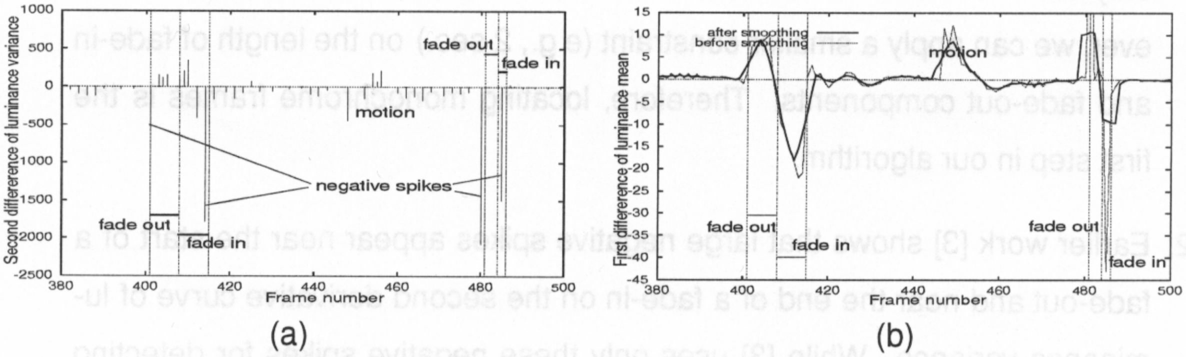
We start with the production model of a fade as described in [3]. Fade-in and fade-out often occur together as a *fade group*. More specifically, a fade group starts with a shot fading out to a color $\mathcal{C}$ which is then followed by a sequence of monochrome frames of color $\mathcal{C}$, and it ends with a shot fading in from color $\mathcal{C}$. Fade groups formed this way are often referred to as a single fade. Alattar [3] detects fades by recording all negative spikes in the second derivative of frame luminance variance curve and ensuring that the first derivative of luminance mean curve is relatively constant next to a negative spike. [7] proposes detecting fades by fitting a regression line on the frame standard deviation curve. We present further extensions to these techniques.

During fade transitions, as detailed in [6], frame luminance mean values move linearly towards $\mathcal{C}$, the fading color, while variance curves of fade-out and fade-in frame sequences have a half-parabolic shape independent of $\mathcal{C}$. The salient features of our enhanced two-step fade detection algorithm are the following:

1. Existence of monochrome frames is a very good clue for detecting all potential fades, and they are used in our algorithm as the first step in recognizing the existence of a fade. In a quick fade, the monochrome sequence may last

only one frame while in a slower fade it would last up to 100 frames. However, we can apply a smaller constraint (e.g., 2 sec.) on the length of fade-in and fade-out components. Therefore, locating monochrome frames is the first step in our algorithm.

2. Earlier work [3] shows that large negative spikes appear near the start of a fade-out and near the end of a fade-in on the second derivative curve of luminance variance. While [3] uses only these negative spikes for detecting dissolves, we observe that motion also would cause such spikes. It can be seen from our simulations shown in Figure 1(a) that two relatively large negative spikes are actually present at the end of the fade-in (near frame 420), and only the second spike corresponds to a real boundary. Therefore, for robustness we should search for all spikes near a monochrome sequence until conditions discussed in the next step (3) are not satisfied. This forms the second step of our algorithm.
3. The first derivative of mean remains relatively constant and does not change its sign during a fade-out or a fade-in (see Figure 1(b)). Since in real videos, the mean feature would be distorted by motion, some smoothing operation needs be applied to the mean difference curve before examining *the constancy of its sign* within a potential fade region. Similarly, depending on whether it is a fade-in or a fade-out, the variance of fading frames will increase or decrease rapidly. This establishes another constraint for testing the existence of fades in our algorithm.
4. We also constrain the variance of the starting frame of a fade-out and the ending frame of a fade-in to be above a threshold to eliminate false positives caused by dark scenes, thus preventing them from being considered as monochrome frames.



(a)

(b)

Figure 1: Fade: (a) The second derivative of luminance variance values during a fade; (b) the first derivative of luminance mean values.

3 Enhancements to Dissolve Detection

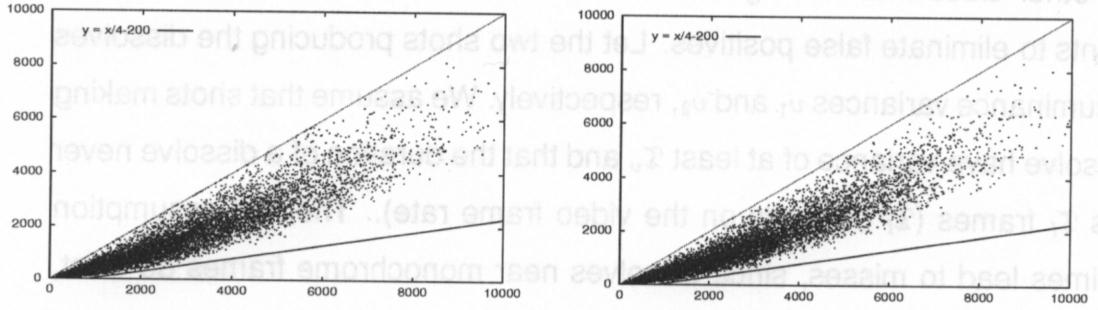
A dissolve is a combination of a fade out and fade in, superimposed on the same film strip [5]. A dissolve occurs when one whole picture fades away while another whole picture appears. It provides a smooth restful transition, with its speed affecting the overall mood and flow of video sequences. Dissolves are often used in dance and music pieces and in some transitions in drama. It is also used in live sports to separate slow motion replays from the live action.

Our approach to dissolve detection is based again on the production model of an ideal dissolve. A fade can be considered as a special dissolve where one shot is composed of monochrome frames. Therefore, similar to a fade, during a dissolve transition, the luminance mean curve changes in a linear fashion, while the luminance variance curve has a parabolic shape. This means the first order difference of the mean curve should be constant during a dissolve , while that of the variance curve should change in a linear fashion. If we take the second order difference of the luminance variance curve, two large negative spikes should appear at the start and end of the dissolve similar to the case of fade transitions. In fact, this feature is the basis for dissolve detection approach proposed by [2]. However, our study of real dissolves suggests that these negative spikes are not easily obvious during dissolves when compared to fades due to noise and motion. Therefore, we ignore these negative spikes in our dissolve algorithm. Instead, we look for other clues that can signal the existence of a dissolve, and for various constraints to eliminate false positives. Let the two shots producing the dissolves be with luminance variances v_1 and v_2, respectively. We assume that shots making up a dissolve have variance of at least $\mathfrak{T}_v$ and that the duration of a dissolve never exceeds $\mathfrak{T}_l$ frames ($\mathfrak{T}_l$ depends on the video frame rate). The first assumption can at times lead to misses, since dissolves near monochrome frames do exist, albeit relatively uncommon. The second assumption is very reasonable, since it is unlikely to have a dissolve lasting longer than 2 seconds. Based on these assumptions, the following steps form the basis of our algorithm.

1. Developing further the mathematical model of a dissolve starting at frame s and ending at frame e presented in [2], we algebraically establish that the first order difference t_i^v of the variance curve changes linearly from a negative value of $-\frac{2(e-s)-1}{(e-s)^2}v_1$ ($< -\frac{2\mathfrak{T}_v-1}{\mathfrak{T}_l}$) at frame s to a positive value of $\frac{2(e-s)-1}{(e-s)^2}$ ($> \frac{2\mathfrak{T}_v-1}{\mathfrak{T}_l}$) at frame $e-1$. Therefore, the existence of all dissolves can be triggered by all zero crossing sequences in the t_i^v curve whose start value is below a negative threshold, which then continuously increases, and then the end value is above a positive threshold. In the actual implementation, to reduce the effect of noise and motion we smooth out the curve before searching for these zero crossing sequences.

2. Due to the smoothing operation, the position of the negative and positive peaks of the t_i^v curve caused by a dissolve is no longer coincident with its actual position in the ideal case. We can adjust these positions by moving the position of the negative peak backward until the value of t_i^v increases beyond a negative threshold. Similarly, the position of the positive peak is moved forward until the value of t_i^v drops below a positive threshold.

3. The variance curve f_i^v has a parabolic shape during a dissolve. It obtains the minimum value of $\frac{v_1 v_2}{v_1+v_2}$ at frame number $\eta = \frac{ev_1+sv_2}{v_1+v_2}$ (we ignore the fact that



(a)

(b)

Figure 2: Hints for threshold selection in dissolve detection: (a) Plot of $\frac{v_1^2}{v_1+v_2}$ against v_1; (b) Plot of $\frac{v_2^2}{v_1+v_2}$ against v_2.

frame number must be an integer). From this, we have:

$$f_s^v - f_\eta^v = \frac{v_1^2}{v_1+v_2} \text{ and } f_e^v - f_\eta^v = \frac{v_2^2}{v_1+v_2}. \tag{1}$$

Figure 2(a) simulates the the plotting of $\frac{v_1^2}{v_1+v_2}$ against v_1. This and the plot of the second part of this equation (Figure 2(b)) together show that all points lie below a line, say $y = x/4 - 200$; therefore, the difference between the start frame and the middle frame of a dissolve should be greater than $\frac{v_1}{4} - 200$. The same condition applies to the difference between the end frame and middle frame of a dissolve. In addition, we have:

$$f_s^v + f_e^v - 2f_\eta^v = \frac{v_1^2 + v_2^2}{v_1 + v_2} > \frac{v_1 + v_2}{2}. \tag{2}$$

By now we already have a set of constraints on the shape of the parabolic curve to eliminate false positives caused by motion. However, these conditions only guide us to set appropriate thresholds. In order to cope with the effects of noise and motion, in the implementation we use lower thresholds.

4 False Positive Transition Elimination

Excluding complex graphics transition effects, changes in lighting, noise, object and camera motion are sources for false detection with shot transition detection algorithms. While those caused by long and fast camera operation can be eliminated by performing a "motion transition removal" test [4, 8], we propose a simple method based on color histogram difference for eliminating other kinds of false positives. Frames belonging to the same shot should be similar if there is no transition in the scene space, i.e., due to camera movements. Once fades and dissolves are detected, for each declared transition, we examine the shot preceding the transition and the shot succeeding it. This transition is declared to be a false positive if the difference between an arbitrary frame from the first shot and some arbitrary frame from the second shot is less than an empirically determined threshold, and is eliminated. This technique can effectively prevent common effects such as flash lights, close-up objects moving in front of the camera, key-in, and other momentary noise.

5 Experimental Results

In order to test our enhanced fade and dissolve detection algorithms on a diverse data set, we collected around 8 hours of video data from TV programs of news, commercials, sports, music, and cartoons telecast on different channels on different days and at different times, and encoded them in the MPEG-1 format. Some clips were recorded more than 5 years ago. A portion of this set (about 1/1/2 to 2hrs of data) was used to test the proposed algorithms and it contained a total of 1373 cuts, 297 dissolve transitions, and 111 fades. We compared the performance of our algorithms with WebFlix, a commercial tool available for editing MPEG videos and also with an implementation of a simple version of the twin-comparison technique [8]. Instead of fine tuning the latter [8], we employed 5 pairs of thresholds, and for each test video and for each type of transition, we chose its best result and used it for comparison.

Our algorithm for detecting dissolves performs better than WebFlix and the twin-comparison approach. It obtains a reasonable good level of recall of around 82%, while that for twin-comparison is 76%. This suggests that our algorithm would be able to detect those dissolves whose color histogram differences between two consecutive frames is small, and which are not detected by twin-comparison. The accuracy of our algorithm is also much better than twin-comparison, since most of the false positives are eliminated by different thresholds set on mean and variance curves. The performance of WebFlix in detecting dissolves is quite poor, as it misses nearly 50% of dissolves while only around 25% of its dissolves is correct. The performance of our fade detection algorithm is very good. Overall, it can detect 93% of fades, and 90% of declared fades are correct. In addition, the algorithm obtains a very high score of 97% and 99% in cover-precision and cover-recall [6], respectively. The lowest performance of our fade detection algorithm is on news videos and it obtains a precision level of only 63%. However, this only slightly affects the overall results, since fades are uncommon in newscasts [6].

6 Conclusion

In this paper we have presented our improved algorithms for detecting different types of shot transition effects such as fades and dissolves. Based on the mathematical models for producing ideal fades and dissolves, different clues (e.g., monochrome frames) for discovering the existence of these effects are proposed, and constraints on the characteristics of frame luminance mean and variance curves are derived analytically in our approach to eliminate false positives caused by camera and object motion during gradual transitions. We also present an effective technique for eliminating false positives from a list of detected transitions. We evaluate our algorithms against two other methods for shot transition detection, WebFlix and twin-comparison, on a variety of videos and demonstrate the better performance of our techniques in terms of recall, precision, cover-recall, and cover-precision.

References

[1] A. Hampapur, R. Jain, and T. Weymouth, “Digital video segmentation,” in *Proceedings of the Second ACM International Conference on Multimedia (MULTIMEDIA ’94)*, (New York), pp. 357–364, ACM Press, Oct. 1994.

[2] A. M. Alattar, “Detecting and compressing dissolve regions in video sequences with a DVI multimedia image compression algorithm,” in *Proceedings of 1993 IEEE International Symposium on Circuit and Systems*, pp. 13–16, 1993.

[3] A. M. Alattar, “Detecting fade regions in uncompressed video sequences,” in *Proceedings of 1997 IEEE International Conference on Acoustics Speech and Signal Processing*, pp. 3025–3028, 1997.

[4] V. Kobla, D. DeMenthon, and D. Doermann, “Special effect edit detection using Video Trails: a comparison with existing techniques,” in *Proceedings of SPIE conference on Storage and Retrieval for Image and Video Databases VII*, Jan. 1999.

[5] D. Arijon, *Grammar of the film language*. Silman-James Press, 1976.

[6] B. T. Truong, “Video genre classification based on shot segmentation.” Honours Thesis, Curtin University of Technology, Western Australia, November 1999.

[7] R. Lienhart, “Comparison of automatic shot boundary detection algorithms,” in *Proceedings of SPIE, Image and Video Processing VII*, vol. SPIE 3656-29, 1999.

[8] H. Zhang, A. Kankanhalli, and S. Smoliar, “Automatic partitioning of full-motion video,” *Multimedia System*, vol. 1, pp. 10–28, 1993.