A System for Reliable Dissolve Detection in Videos

Rainer Lienhart and Andre Zaccarin
Microprocessor Research Labs, Intel Corporation, Santa Clara, CA 95052, USA
{Rainer.Lienhart, Andre.Zaccarin}@intel.com

Abstract

Automatic shot boundary detection has been an active research area for nearly a decade and has led to high performance detection algorithms for hard cuts, fades and wipes. Reliable dissolve detection, however, is still an unsolved problem. In this paper, we present the first robust and reliable dissolve detection system. A detection rate of 75% was achieved while reducing the false alarm rate to an acceptable level of 16% on a test video set for which so far the best reported detection and false alarm rate had been 66% and 59%, respectively. In addition, a dissolve's temporal extent is estimated, too. The core ideas of our novel approach are firstly the creation of a dissolve synthesizer capable of creating in principle an infinite number of dissolve examples of any duration from a video database of raw video footage allowing us to use advanced machine learning algorithm such as neural networks and support vector machines which require large training sets, secondly, two simple features capturing the characteristics of dissolves, thirdly, a fully temporal multi-resolution search based on a fixed-position and fixed-scale transition/special effect detector enabling us to determine also the true duration of detected dissolves, and finally, a post-processing step which uses global motion estimation to further reduce the number of falsly detected dissolves.

1 Introduction

Almost all current shot detection methods are either based on simple rules or simple statistical tests. While such 'simple' approaches have been quite successful for hard cut, fade and wipe detection, all approaches proposed so far have problems with detecting dissolves reliably. In this paper we therefore propose to apply advanced pattern recognition and machine learning techniques, which have proven themselves to be suitable for complex detection and recognition tasks such as face detection, to the problem of reliable dissolve detection [5,7].

There might be a good reason why people have never tried this before for complex transitions such as dissolves, wipes and swirls: It is tedious to hand-label training examples, especially if they do not appear as regular and often as hard cuts. We will argue in Section 4 that a properly designed dissolve synthetizer makes the process of tedious hand-labeling superfluous. Obviously the synthesizer approach is not restricted to dissolves, but can be applied to any transition effect. Therefore, we will often use the term 'transition' instead of 'dissolve' to point this out. Section 5 introduces two computationally cheap features to capture the characteristics of dissolves, before Section 6 details the training and classification process. It is followed by detailed experimental results. Section 8 concludes the paper.

2 Related Work

Much work has been done on automatic shot boundary detection in videos. Early work concentrates mainly on hard cuts, while summarizing all other kinds of transitions undifferentiatedly under "soft" cuts. Reported performance numbers were usually dominated by hard cuts [1,3,4,9]. Recent related work geared towards specialized detectors. For instance [8] addresses the problem of wipe detection.

So far, no technique for reliable detection of transition effects beyond hard cuts, fades, and wipes has been published. Although there are a few techniques for dissolve detection with a sufficient hit rate between 50% and 80%, reported false alarm rates of 100% and up clearly identifies them as unreliable [4].

Dissolves are inherently difficult to detect since two video sequences are temporally as well as spatially intermingled at any time. In order to employ a dissolve's definition directly for detection, the two sequence must be separated. Unfortunately, the separation of two intermingled sources is inherently difficult.

3 Design Decisions and System Overview

DEFINITION. A dissolve sequence $D(\mathbf{x},t)$ is defined as the mixture of two video sequences $S_1(\mathbf{x}, t)$ and $S_2(\mathbf{x}, t)$, where the first sequence is fading out while the second is fading in:

$$D(\mathbf{x}, t) = f_1(t) \cdot S_1(\mathbf{x}, t) + f_2(t) \cdot S_2(\mathbf{x}, t) \ , t \in [0, T] \qquad (1)$$

The most common dissolve types are cross-dissolves with

$$f_1(t) = (T-t)/T \ , f_2(t) = t/T \qquad (2)$$

and additive dissolves with

$$f_1(t) = \begin{cases} 1 & \text{if } (t \le c_1) \\ (T-t)/(T-c_1) & else \end{cases} , \ f_2(t) = \begin{cases} t/c_2 & \text{if } (t \le c_2) \\ 1 & else \end{cases}$$

$$c_1 = \ ]0, T[\ \ , c_2 = \ ]0, T[\qquad (3)$$

TYPES OF DISSOLVES. Basically three different kinds of dissolves can be distinguished based on the visual difference between the two shots involved:

(1) The two shots involved have different color distributions. Thus, they are different enough such that a hard cut would be detected between them if the dissolve sequence were removed.
(2) The two shots involved have similar color distributions which a color histogram-based hard cut detection algorithm would not detect, however, the structure between the images is different enough in order to be detected by an edge-based algorithm.
(3) The two shots involved have similar color distributions and similar spatial layout. This type of dissolve represents a special type of morphing.

DESIGN DECISIONS. In this work, we concentrate only on the first two types of dissolves, since they clearly mark transitions between semantic shots. The morphing-like dissolves are ignored since they do only represent a transition from a technical point of view and not from a semantic point of view. Moreover, we restrict our detection scheme to dissolves lasting between 0.2 and 3

0-7803-6725-1/01/$10.00 ©2001 IEEE

seconds, though our system can handle any range of durations.

SYSTEM OVERVIEW. The overall system consists of two large components:

(1) At the core of the **training system** is the transition synthesizer. The transition synthesizer can create from a proper video database an infinite number of dissolve examples. We use it to create a large training and validation set of dissolves with a fixed length and a fixed position of the dissolve center. These sets are then used to train iteratively with the so-called bootstrap method a heuristically optimal fixed-scale and fixed-position transition detector [6].

(2) A **multi-resolution transition detection approach** is then used to detect transition. In a first step, various frame-based features are derived (Figure 1(a)). Each frame-based feature forms a time series, which in turn is re-scaled to a full set of time series at different sampling rates creating a time series pyramid (Figure 1(b)). At each scale, a fixed-size sliding window runs over the time series, serving as the input to a fixed-scale and fixed-position transition detector (Figure 1(c)). The fixed-scale and fixed position transition detector outputs the probability that the feature sequence in the window was produced by a transition effect. This results in a set of time series of transition effect probabilities at the various scales (Figure 1(d)). For scale integration, all probability times series are rescaled to the original time scale (Figure 1(e)), and then integrated into a final answer about the probability of a transition at a certain location and its temporal extent (Figure 1(f)).

4 Transition Synthesizer System

The **VIDEO DATABASE** serves as the source of video sequences for the transition synthesizer. It should consist of a diverse set of videos. In the ideal case, all videos in the database are annotated by their transition free video sub-sequences (henceforth called shots). In order to create the transition example this information is essential for the transition synthesizer to avoid accidentally using two video sequences that already contain other transition effects.

The ideal video database can be approximated by adding only videos to the database for which transitions besides hard cuts and fades are rare. Current state of the art shot detection algorithms can perform hard cut and fade detection reliably. We used the hard cut algorithm proposed in [9] and the fade detection algorithm proposed in [4] to automatically pre-segment 7 hours of home videos without errors.

The **TRANSITION SYNTHESIZER** is supposed to generate a random video sequence containing the specified number of transition effects of the specified kind. The following parameters must be given before the synthetic transitions can be created:

- N = Number of transition to be generated
- $P_{TD}(t)$ = Probability distribution of the durations of the transition effect
- R_f, R_b = Amount of forward and backward run before and after the transition. Usually, R_f and R_b will be set to the same value.

Then the transition synthesizer works as follows:

(1) Read in the list of all videos in the database together with their shot description.

(2) For i = 1 to N

  (2.1) Randomly choose the duration d of the transitions according to $P_{TD}(t)$

  (2.2) Determine the minimal required duration for both shots as $(d + R_f)$ and $(d + R_b)$, respectively.

  (2.3) Randomly choose both shots $S1=[t_{s1},t_{e1}]$ and $S2=[t_{s2},t_{e2}]$ subject to their minimal required duration.

  (2.4) Randomly select the start time t_{start1} and t_{start2} of the transition for $S1$ and $S2$ subject to $t_{s1}+R_f < t_{start1} < t_{e1} - d$ and $t_{s2} < t_{start2} < t_{e2} - R_b - d$.

  (2.5) Create the video sequence as $S1(t_{start1}-R_f,t_{start1})$ + Transiton($S1(t_{start1},t_{start1}+d),S2(t_{start2},t_{start2}+d)$) + $S2(t_{start2}+d,t_{start2}+d+R_b)$

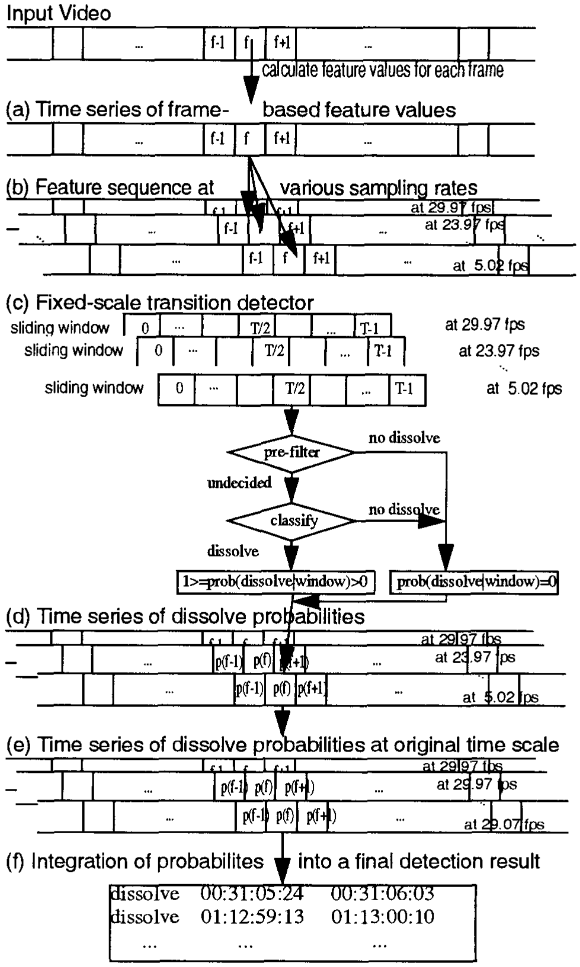

Input Video
f-1
f
f+1
...
calculate feature values for each frame
(a) Time series of frame- based feature values
(b) Feature sequence at various sampling rates
at 29.97 fps
at 23.97 fps
at 5.02 fps
(c) Fixed-scale transition detector
sliding window
0
T/2
T-1
at 29.97 fps
at 23.97 fps
at 5.02 fps
pre-filter
no dissolve
undecided
classify
no dissolve
dissolve
1>=prob(dissolve|window)>0
prob(dissolve|window)=0
(d) Time series of dissolve probabilities
p(f-1)
p(f)
p(f+1)
(e) Time series of dissolve probabilities at original time scale
(f) Integration of probabilites into a final detection result
dissolve 00:31:05:24 00:31:06:03
dissolve 01:12:59:13 01:13:00:10
... ...

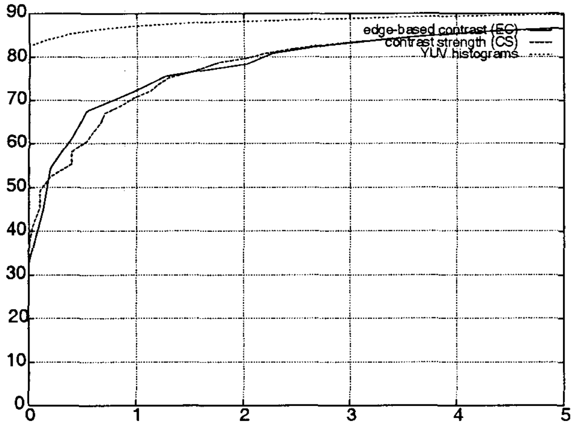
Figure 1. System overview of the transition detection system

5 Dissolve Features

In this section we discuss two different classes of features: contrast-based and color-based features. Both types of features are influenced by dissolves in the same way, however, they respond sometimes differently to typical false alarm situations. Thus using both kinds of features jointly reduces the false alarm rate.

CONTRAST-BASED FEATURES. Generally, the image contrast decreases towards the center of a dissolve and recovers as the dissolve ends. This characteristic pattern can be captured by the time series of the average contrast strength of each frame. A fast measure of the contrast strength is the sum of the magnitude of the directional gradients:

Figure 2. Performance of the three features for pre-filtering.



$$CS_{avg}(t) = \left(\sum_{x \in X} \sum_{y \in Y} \left| \frac{\partial}{\partial x} I(x, y, t) \right| + \left| \frac{\partial}{\partial y} I(x, y, t) \right| \right) / (|X||Y|) .(4)$$

The contrast-based feature strongly responds to type 2 and most type 1 dissolves, since a different color content in the respective shots usually goes along with different structural content.

For comparison, we also used the Edge-based Contrast (EC) introduced in [4]. The *EC* captures and amplifies the relation between stronger and weaker edges and is based on the Canny edge detector. In our experiments both features were almost always perfectly correlated, however, the *CS* is much faster to compute and was therefore used in our experiments.

COLOR-BASED FEATURES. Based on the taxonomy of dissolve types, typical color-based features should only be able to capture type 1 dissolves. Fortunately, nearly all type 1 dissolves are also type 2 dissolves, and most type 2 dissolves are also type 1 dissolves. We use a 24 bin YUV image histogram (8 bin per channel) to capture the temporal development of the color content.

6 Training and Classification

The frame-based features introduced above show off a characteristic pattern during a dissolve. It is our goal to develop a fixed location and (almost) fixed duration dissolve classifier, which can plug-in into our multi-resolution detection approach (Figure 1). In addition, a simple and fast pre-filter is designed, too. The main purpose of the pre-filter besides reducing the computational load is to restrict the training samples to the positive examples and those negative examples, which are more difficult to classify. Such a focused training set usually improves the classification performance.

PRE-FILTERING. The time series of our dissolve features almost always exhibit a flat graph. Exceptions are sections with transitions, camera motion and/or object motion. Thus, the difference between the largest and smallest feature value in a small input window centered around the location of interest is used for pre-filtering. If the difference is less than a certain empirical threshold the location will be classified as non-dissolve and not further evaluated. For multi-dimensional data, the maximum difference between the maximum and minimum in each dimension is used as the criterion. The input window size was set empirically to 16 frames.

Figure 2 shows the percentage of falsely discarded dissolve location (x-axes) versus the percentage of discarded locations (y-axes). The data has been derived from our large training video set. As can be seen from Figure 2, the YUV histograms outperformed the other features. Combining YUV histograms with CS by a simple OR strategy (one of them has to reject the pattern), performed even better, and was chosen as the pre-filter. The missed rate of accidentally discarded dissolve locations were set to 2% in all the subsequent experiments.

FIXED-SCALE/POSITION TRANSITION DETECTOR. Given a 16-tap input vector from the time series of feature values, the fixed scale transition detector is supposed to classify whether the input vector is likely to be calculated from a certain type of transition lasting about 16 frames. There exist many different techniques for developing a classifier. For our work, we used a real-valued feed-forward neural network with (NN) hyperbolic tangent activation function. The size of the hidden layer was 8, which in turn were aggregated into one output neuron. The value of the output neuron was interpreted as the likelihood that the input pattern has been caused by a dissolve.

For training and validation, we synthesized each 10 hours of dissolve videos with 1000 dissolves, each lasting 16 frames. The four 16-tap feature vectors around each dissolve's center were chosen to form the dissolve pattern training/validation set. All other patterns which did not overlap with a dissolve and which were not discarded by the pre-filter formed the non-dissolve training/validation set. Thus, each training and validation set contained 4000 dissolve examples, and about 20000 non-dissolve examples.

Initially 1000 dissolve patterns and 1000 non-dissolve patterns were selected randomly for training. Only the non-dissolve pattern set was allowed to grow by means of the so-called 'bootstrap' method. This method, proposed by Sung [6], starts with an initial set of non-dissolve patterns to train the NN. Then, the trained NN is evaluated using the full training set. Some of the falsely classified patterns of the training set are randomly added to the non-dissolve set and a new, hopefully enhanced NN is trained with the extended pattern set. The resulting NN is evaluated with the training set again and additional falsely classified non-dissolve patterns are added to the set. This cycle of training and adding new patterns was repeated until the number of falsely classified patterns in the validation set did not decrease anymore or nine cycles had been evaluated. Usually between 1500 and 2000 non-dissolve pattern were added to the actual training set by this procedure. The NN with the best performance on the validation set was selected for classification.

SCALE INTEGRATION. In our experiments, we observed that the fixed-scale and fixed-position transition detector could be very selective. It might only respond to a dissolve at one scale. Therefore, we choose to implement a winner-takes-all strategy: If two detected dissolve sequences overlap, then the one with the highest probability value wins, i.e., the other is discarded. The competition starts at the smallest scale (shortest detected dissolve candidates) competing with the second smallest scale and goes up incrementally to the largest scale (longest detected dissolve candidates). This approach has been proven to be very effective in determining the right duration of a dissolve.

POST-FILTERING. A large fraction of the remaining false alarms are caused by camera motion such as pans. We have implemented a 4-parameter global motion estimation algorithm [2] and ran this motion estimation on the intervals where we have detected dissolves. We then processed the raw results to decide if global motion is present within each interval. For each frame, we accepted the global motion parameters computed for that frame if they were consistent with the previous and following 2 frames, and if the translational motion was larger than one pixel. If motion parameters were accepted for more than 2/3 of the frames in a given interval, we concluded that there was global motion and rejected the dissolve hypothesis.

Table 2 shows the results with and without taking into account the global motion estimation. As can be seen, taking into account camera motion reduces the number of false alarms. For some cases, however, some true dissolves are incorrectly removed. This situation arises when the camera motion in both segment of the dissolve is similar, or when the center of the detected dissolve is off the true dissolve center. This last situation is typically caused by camera motion on one of the dissolve segment which interferes with the detection algorithm. To compensate for this problem, we need to find an approach to integrate the camera motion parameters with the dissolve detection algorithm instead of using it as a post-processing tool as we currently do.

7 Experimental Results

VIDEO SETS. The training video database was composed of 7 hours of home videos. All videos were encoded as MPEG-1 and contained only hard cuts none of which were missed or incorrectly detected by our hard cut detection algorithm [9]. The database was used to create the training and validation pattern sets for the NN. Both pattern sets were derived from 10 hours of synthetic video with 1000 dissolve each lasting 16 frames. The training patterns were also used to derive appropriate thresholds for the prefilter.

Dissolve detection performance was measured on 5 different video sequences lasting about 3.5 hours (see Table 1). Four of them have already been used in the comparative study of shot detection algorithms in [4] enabling comparison with two recently proposed dissolve detection approaches. Since those four sequences are not publicly available, a newscast called 'News1' from the MPEG-7 test set was added to enable comparison with other dissolve detection techniques in future.

Table 1: List of test video sequences

Video	News1	Bay-watch	Heute	Ground-hog Day	Dissolves	Σ
Duration (hh:mm:ss)	38:18	50:42	10:35	1:34:35	16:50	3:20:25
# hard cuts	297	976	78	773	140	2264
# fades	1	19	1	7	12	40
# dissolves	24	100	2	6	276	408
# wipes	9	-	2	-	-	11
# shots	332	1096	84	787	429	2728

COMPARISON PROCEDURE. Given the total number of dissolves, their locations and durations, the performance of the different algorithms are measured by:

- **hit rate** *h* which is the ratio of correctly detected dissolves to its actual number of dissolves
- **false hits f** which is the ratio of falsely detected dissolves to the actual number of dissolves

We count each detected dissolve as a hit if and only if it temporally overlapped at least by 40% with an actual dissolve. Multiple detections of the same dissolve were counted only once and occurred only one time during our tests.

PERFORMANCE. One of the biggest advantage of our approach is that it requires only the specification of two easily understandable parameters:

(1) The percentage of falsely rejected dissolves by the pre-filter which was set to 2% on the training set in experiments and

(2) The minimal required response of the NN in order to declare a dissolve. This threshold was set to 0.75 in experiments.

As can be seen in Table 2, our novel approach outperformed the approaches based on the edge change ratio and edge contrast [4,10]. At the same or better hit rate, the false alarm rate was always significantly smaller. Note also that on a qualitative basis we observed that the detector determined reliably the extent of dissolve transitions.

Video		Our Approach		Edge Contrast		ECR	
		hits	false hits	hits	false hits	hits	false hits
News1	#	20 / 18	14 / 9	-	-	-	-
	%	83/75%	58/37%				
Ground-hog Day	#	5 / 5	22 / 13	1	24	4	1113
	%	83.3%	366/217%	16.7%	400%	66.7%	37100%
Heute	#	1 / 1	1 / 0	2	3	0	110
	%	50%	50/0%	100%	150%	0%	5500%
Bay-watch	#	70 / 64	21 / 9	55	184	67	715
	%	70/64%	21/9%	55%	184%	67%	715%
Dis-solves	#	208/208	8 / 4	155	28	198	135
	%	75%	2.9/1.4%	56%	10%	72%	49%
Σ		304/296	66 / 35	213	239	269	2073
		75/73%	16/8.5%	52%	59%	66%	508%

Table 2: Comparison between our dissolve detection approach and the approaches based on edge change ratio (ECR) [10] and edge contrast [4]

8 Conclusion and Outlook

Reliable dissolve detection is an inherently difficult problem. However, our approach achieved outstanding performance compared to the approaches based on the edge change ratio and edge contrast. At the same time, our approach could determine very well the extent of a dissolve.

9 References

[1] A. Dailianas, R. B. Allen, P. England: Comparison of Automatic Video Segmentation Algorithms. In Proc. SPIE 2615, pp. 2-16, Oct. 1995.

[2] F. Dufaux and J. Konrad. Efficient, Robust and Fast Global Motion Estimation for Video Coding. IEEE Trans. on Image Processing, vol. 9, no. 3, March 2000, pp. 497-501.

[3] U. Gargi, R. Kasturi, S. H. Strayer. Performance Characterization of Video-Shot-Change Detection Methods. IEEE Trans. on Circuits and Systems for Video Technology, Vol. 10, No. 1, Feb. 2000.

[4] R. Lienhart. Comparison of Automatic Shot Boundary Detection Algorithms. In Proc. SPIE 3656-29, Jan. 1999.

[5] A. Wernicke and R. Lienhart. On the Segmentation of Text in Videos. IEEE Int. Conf. on Multimedia and Expo, Vol. 3, pp. 1511 -1514, July 2000.

[6] K.-K. Sung. Learning and Example Selection for Object and Pattern Detection. PhD Thesis, MIT AI Lab, Jan. 1996. Available as AI Technical Report 1572.

[7] H. A. Rowley, S. Baluja, and T. Kanade. Neural Network-Based Face Detection. IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 20, No. 1, pp. 23-38, Jan. 1998.

[8] M. Wu, W. Wolf, and B. Liu. An Algorithm for Wipe Detection. ICIP 98 , vol.1, pp. 893 -897, 1998.

[9] B. Yeo and B. Liu. Rapid Scene Analysis on Compressed Video. IEEE Trans. on Circuit and Systems for Video Technology, Vol. 5, No. 6, Dec. 1993.

[10]R. Zabih, J. Miller, and K. Mai. A Feature-Based Algorithm for Detecting and Classifying Scene Breaks. Proc. ACM Multimedia 95, San Francisco, CA, pp. 189-200, Nov. 1995.