ANALYSIS-BY-SYNTHESIS DISSOLVE DETECTION

Michele Covell *Subutai Ahmad*
covell@ieee.org *sahmad@yesvideo.com*
YesVideo, 2249 Zanker Road, San Jose, CA 95131

ABSTRACT

This paper presents a novel, real-time, minimal-latency technique for dissolve detection which handles the widely varying camera techniques, expertise, and overall video quality seen in amateur, semi-professional, and professional video footage. We achieve 88% recall and 93% precision for dissolve detection. In contrast, on the same data set, at a similar recall rate (87%), DCD has more than 3 times the number of false positives, giving a precision of only 81% for dissolve detection.

1. OVERVIEW

This paper discusses an improved approach for dissolve detection. A *dissolve* gradually cross-fades from the old shot's footage to the new shot's footage. The dissolve is the most common transition used in post-production. It is also available as an "in-camera" effect on many consumer-grade camcorders. We use the results from our dissolve detector (along with our cut and fade detectors) to support scene-based video browsing and editing [1]. By placing our detector at the heart of an inexpensive consumer product, we have been forced to make it both computationally efficient and robust to the widely varying camera techniques, expertise, and video quality seen in amateur and semi-professional footage.

This paper does not describe our approach to cut or fade detection due to the extensive and successful prior art [2,3]. Instead, after a short introduction to dissolve detection (Section 2), we describe our new approach (Section 3). Section 4 presents our precision and recall for dissolve detection and compares these to the results we get using the best published approach. Section 5 concludes by summarizing our approach.

2. BACKGROUND

Many approaches to dissolve detection have been published over the years. The published approaches to dissolve detection fall into 3 broad categories.

The first category, *temporal pattern matching on edge-based statistics*, detects edges, creates a single summary statistic describing the edges in each frame, and then matches the evolution of these summary statistics over time against fixed, characteristic patterns for dissolves. One example in this category is the *edge change ratio*. With motion compensation to allow edge tracking, Zabih [4] uses the percentages of edges that appear or disappear between a pair of frames as his summary statistic. Another example in this category is the *edge-based contrast*: Lienhart [5] uses the relative number of "strong" to "weak" edges within each frame as his summary statistic. For recall levels[1] around 60-70%, the precision of these approaches is between 8% and 38% [5].

The second category, *temporal pattern matching on pixel-level statistics*, takes a similar pattern-matching approach. Instead of using statistics derived from edge detection, it uses simple pixel-level statistics. A common low-level statistic is the within-frame intensity mean and standard deviation [3,6]. For dissolve detection, the time course of the variance is typically matched to a parabola, and the mean is matched to a line. Using this approach, Truong [3] reports equal recall–precision rates of below 65% on the dissolves in news-program footage.

The final category of dissolve detection is *temporal pattern matching using synthetic-dissolve statistics*. Our approach, *analysis-by-synthesis*, falls in this category. Our premise is that simple pixel- and edge-based pattern matching are inadequate measures of the probability of a dissolve, since the patterns that we are trying to match are too closely tied to the specific footage combined in the dissolve. Instead, we predict the appearance of a dissolve between segments of footage and then compare the observed footage to the predicted appearance.

The only published approach in this category is the double chromatic difference (DCD) [7,8]. The first step of the DCD segments the video into non-overlapping categories of "potential dissolves" and "non-dissolves" using edge-based [7] or pixel-level [8] statistics, as described above. The second step of the DCD detector uses this segmentation to define one synthetic dissolve per potential-dissolve segment, beginning and ending at the first and last frame of the segment, respectively. From these starting and ending frames, the center frame of a synthetic dissolve is formed and compared to the

[1] Recall = TP/(TP+FN) and precision = TP/(TP+FP) where TP=true positives, FP=false positives, and. FN=false negatives.

intervening footage. If the shape of the comparison error over time is bowl shaped, the potential-dissolve segment is accepted.

There are shortcomings with the DCD formulation. The first step of the DCD eliminates the vast majority of the possible dissolves (most dissolve positions and lengths), based on an inadequate measure. The second step of the DCD, with the more powerful testing procedure, is presented with an extremely small set of potential dissolves to simply accept or reject. Furthermore, there is no easy set of thresholds that can be used to shift the preponderance of the solution from the first step to the second step. A high threshold in the first step results in too many frames being eliminated as part of non-dissolve segments. A low threshold results in true dissolves being buried in the middle of much longer segments, resulting in a synthetic dissolve unlike the buried dissolve. Finally, even the second DCD step does not use the synthetic dissolve to full advantage. It relies on matches against a single frame from that dissolve, instead of more robust comparisons using a wider window of synthetic frames. Even with these shortcomings, the DCD performance is well above other published approaches, giving a precision of better than 90% for a recall rate near 70% in our tests.

As we report in Sections 3 and 4, our analysis-by-synthesis approach avoids the shortcomings of the DCD. First, we do not pre-segment the video: instead, we postulate a short dissolve nearly everywhere and move the boundaries of the dissolve based on a combination of synthetic-dissolve tests. Second, we use more of the synthetic dissolve, thereby increasing the robustness of our detection approach.

3. DISSOLVE-DETECTION APPROACH

In this section, we describe our approach to dissolve detection. The core of our detector synthesizes a synthetic dissolve and compares those synthetic frames against the observed footage. We synthesize the synthetic dissolve using a cross fade between the frames at the nominal starting and ending time of the potential dissolve. This implicitly assumes that the component footage is unchanging during the dissolve and that the only source of change is the transition process itself. As we shall see in Section 4, more than half of our false negatives can be traced back to this implicit assumption. We then use the product of non-linear measures of pixel differences and histogram differences [2] to determine how close the observed footage is to the expected dissolve appearance. The final summary statistic is then the ratio of the difference between the dissolve start and end frames to the average difference between the synthetic-dissolve frames and the observed footage.

To avoid estimating the actual start and end indices of the dissolve, we use a comparatively short nominal dissolve length (e.g., 20 frames) and exploit the linear nature of the dissolve transition: using the frames at t_a and t_b to create a synthetic dissolve will accurately model the time evolution of any linear dissolve containing those two frames. We can see this using geometric reasoning. Consider an idealized dissolve that starts and ends at t_s and t_e, where $t_s \le t_a < t_b \le t_e$. Using $\mathbf{I}_s$ to represent the frame at time t_s and so on, we can represent all of the frames of the dissolve as a straight line segment through image space, starting at $\mathbf{I}_s$ and ending at $\mathbf{I}_e$. The frames $\mathbf{I}_a$ and $\mathbf{I}_b$ are part of this straight-line segment and the frames between times t_a and t_b lie on the straight-line segment between frames $\mathbf{I}_a$ and $\mathbf{I}_b$. This observation means that frames from a dissolve are equally well modeled by nominal-length dissolves between intermediate frames. In fact, for long dissolves that include some amount of motion, the intermediate frames are *better* modeled by a sequence of shorter, nominal-length dissolves, since these shorter line segments through image space will tend to be closer to the actual path through that space.

After locating nominal length dissolves that pass our threshold, we merge together the detected dissolves that overlap by more than ½ of the nominal dissolve length. This avoids marking longer-than-nominal-length dissolves more than once.

Figure 1 illustrates our core detector approach to dissolve detection. Note that, even when some of the nominal-dissolve tests incorrectly fail, due to camera flashes and other localized irregularities, the passing nominal-length dissolves on either side of the failing tests are often correctly merged together. This happens twice in the first dissolve shown in Figure 1.

The remainder of this section describes techniques that we use to make the analysis-by-synthesis approach faster than real-time on off-the-shelf processors. First, all of

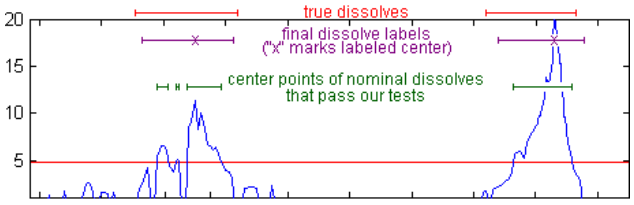

Figure 1: Analysis-by-Synthesis Detection: A nominal-length synthetic dissolve is compared to the observed footage. The summary statistic for each comparison (the solid curve) is the ratio of how well the synthetic dissolve matches the observed footage to how well the start frame matches the end frame. To avoid "degenerate" dissolves, if the start and end frames are too similar, the summary statistic is taken as one. Nominal-length dissolves with a summary statistic above 4.8 pass. Overlapping nominal-length dissolves are merged into a single detection. In this figure, there are 2 true dissolves and 2 detections.

our analyses operate on quarter-sized MPEG1 frames (176x120). This reduces the analysis time by four times.

We also use two simple, inexpensive tests before the synthetic-dissolve creation to reduce the computational load. These simple early-stage tests rely on the general production principle that you should not dissolve between two shots that look the same. This means that the starting, center, and ending frames must all be "sufficiently different" from one another. To enforce this requirement, we compare frames separated by half of a nominal dissolve length. The frames are not sufficiently different from the frames ½ dissolve earlier and later are dropped from the set under consideration. On the frames that do pass, we compare the frames ½ nominal-dissolve-length before and after the passing frame. Only the frames that pass both the half-dissolve and the full dissolve tests are tested as the center location of a nominal-length dissolve. Gating our synthetic-dissolve comparisons in this way typically results in a 23% reduction in computation and changes the recall/precision performance by less than 1%.

Finally, we further reduce the computational load associated with each synthetic dissolve by creating and comparing only the middle 50% of each nominal dissolve. This reduces the computational load of this step by half, while still providing us robustness by comparing many of the frames of the synthetic dissolve. Since the first and last 25% of the synthetic dissolve are very similar to the start and end frames, respectively, these comparisons provide less discrimination than do the center 50% of the synthetic dissolve: comparing only the center 50% gives the same recall/precision to within 1% as comparing the full dissolve length.

With all of these optimizations, our system can run cut, dissolve, and fade detection in 9.8 ms/frame, on average, on a 2 GHz Pentium 4. Of that, an average of 2.9 ms/frame is taken up by MPEG1 decoding.

There are many thresholds within our analysis-by-synthesis dissolve detection and the optimization surface is highly non-linear. This leaves open the issue of how to best set these thresholds. We have taken the approach of setting a few of the basic thresholds (such as the histogram bin size used in histogram comparison and how dissimilar two gray-level pixels must be to count as "different" in the pixel-difference measures [2]) and then tuning the remaining parameters using a brute force, dense sampling approach. We densely sample the domain of the remaining parameters and pick the combination that gives us the best recall and precision on our training set. The time required to densely sample the parameter domain is actually quite small. Since we cache all of the comparisons that we need, computing the *approximate* false positive/false negative rates is a simple a matter of tabulation. This performance estimate can be determined in well under 1 second per combination, over the entire training set, so dense sampling of the parameter surface is neither difficult nor time consuming.

4. RESULTS

In this section, we report our test results on semi-professional footage, with cuts, dissolves, and fades between shots. For comparison, we also report DCD dissolve-detection results on the same database.

Our test set consists of 7 movies, totaling 5 ½ hours of footage, taken by 5 different professional and semi-professional wedding videographers. Even though this footage has been taken by experienced videographers, it includes some under-produced footage: several extremely fast pans (where a cut would have been more appropriate), several places where the scene is completely occluded by someone walking or standing in front of the camera, and a few sections where the videographer forgot the camera was on and taped its swing towards the floor.

In this 5 ½ hours of footage, the test set contains 351 cuts, 834 dissolves, and 16 transitions of other types. We marked the ground-truth start and end frames for all 834 dissolves. We counted false positives and false negatives using detected-center-within-true-dissolve and one-for-one counting.[2] The only exception to this rule is for falsely detected dissolves that are centered at or near a cut: those false detections are not counted, since any real application (including editing-by-browsing [1]) would suppress these dissolves after cut detection. In this paper, since we are not describing our cut-detection algorithm, we remove cuts and cut-related detections from our results.

For comparison, we implemented a simple version of the DCD, as described by Lu [8]. We approximated the segmentation steps by fitting a parabola to the within-frame variance across 7 consecutive frames. When the error in that fit was low enough and the minimum of the parabola was within the range of those 7 basis points, we extended the end points of the fit forwards and backwards in time, until the least-squares error indicated a poor fit. We tuned our thresholds on this step to result in approximately 80% recall and 40% precision. We favored recall over precision since the precision would ultimately increase (through synthetic-dissolve rejections).

[2] If there are no detected dissolves centered within an actual dissolve, that actual dissolve is counted as a false negative. If there is one or more detected dissolves centered within an actual dissolve, the first detected dissolve is counted as a true positive and the remaining detected dissolves are counted as false positives. Finally, if the center of the detected dissolve falls outside of all of the true-dissolve periods, the detected dissolve is counted as a false positive.

Approach	TP	FN	FP	Recall	Pre-cision
analysis-by-synthesis	738	96	54	88%	93%
DCD	588	246	43	71%	93%
single-frame synth. test	722	112	164	87%	81%

Table 1: Recall and precision for our dissolve-detection approach.

Table 1 reports our dissolve-detection results. The first row of the table shows the results for our proposed analysis-by-synthesis detector. The second line shows the results we achieved using our implementation of DCD. For the same precision rate (93%), the recall rate for DCD is significantly lower than for analysis-by-synthesis (71% versus 88% recall).

To determine whether or not the comparison using multiple synthetic-dissolve frames improves the performance of our detector, we also implemented a version of analysis-by-synthesis that, like the last step of the DCD, compares just the center frame of the synthetic dissolve against all the frames in the hypothesized dissolve and then matches that comparison curve to a parabola. We avoided introducing an explicit segmentation step, such as used in the DCD, by testing for dissolves at all indices, using a nominal length dissolve (e.g., 20 frames). Table 1 shows that, for similar recall rates (87%), the precision rate suffers when the synthesized-to-observed comparison uses only one frame of the synthetic dissolve to test against the full length of the hypothesized dissolve footage (81% versus 93%).

The remainder of this section analyzes the dissolve-detection results for analysis-by-synthesis in more detail. We had 54 false positives from our dissolve detector. Of those, 26 could be traced back, at least in part, to the lack of motion compensation. Most of these 26 occurred during a pan, dolly or zoom (or a combination of those). Some included a pan past a blurry, near-field object. The remaining false positives are due to smooth, extended auto-gain or white-balance changes; due to marking a single, very long (3 second) transition twice; or (in one case) due to marking a different type of gradual transition ("de-tiling").[3]

We had 96 false negatives with the analysis-by-synthesis approach. Of those, 58 could be traced back, at least in part, to the lack of motion estimation: these false negatives were on dissolves where one or both of the component segments included a strong camera pan or zoom. The other false negatives were on dissolves between two similarly colored sequences, often between two pure black-and-white or septa-toned sequences or on dissolves that were much shorter than expected (3-5 frames long).

[3] Since the marked "de-tiling" transition was not an actual dissolve and since we are not counting the remaining 15 non-dissolve, gradual transitions as false negatives, we counted this detection as a false positive.

5. SUMMARY

The analysis-by-synthesis dissolve detector outperforms previously reported techniques. Our recall and precision for dissolve detection on consumer and semi-professional footage are all high enough to support our target application of editing-by-browsing software [1].

Based on a detailed analysis of our results, it is clear that including motion estimation in our system would improve our results. More than half of the false positives and false negatives can be traced back, at least in part, to the effects of uncompensated camera motion or zoom. After that, the next step for improvement is a better model of auto-gain adjustments.

REFERENCE

[1] S.-W. Fu. "YesVideo Launches New Videotape-to-CD Transfer Service to Help Consumers Preserve and Share their Stockpiled Home Videos." www.yesvideo.com/app/Company/Press1127002.asp

[2] H.-J. Zhang, et al. "Automatic Partitioning of Full-Motion Video." *Multimedia Systems*, vol. 1, p. 10–28, 1993.

[3] B.-T. Truong, et al. "New Enhancements to Cut, Fade, and Dissolve Detection Processes in Video Segmentation." *Proc. ACM International Conf. Multimedia*, p. 219–227, 2000.

[4] R. Zabih, et al. "A Feature-Based Algorithm for Detecting and Classifying Production Effects." *Multimedia Systems*, vol. 7, p. 119–128, 1999.

[5] R. Lienhart. "Comparison of Automatic Shot Boundary Detection Algorithms." *Proc. SPIE*, vol. 3656, p. 290–301, 1998.

[6] W. Fernando, et al. "Fade and Dissolve Detection in Uncompressed and Compressed Video Sequences." *Proc. ICIP*, vol. 3, p. 299–303, 1999.

[7] H. Yu, G. et al. "Feature-based Hierarchical Video Segmentation." *Proc ICIP*, vol. 2, p. 498–502, 1997.

[8] H. Lu, et al. "Robust Gradual Scene Change Detection." *Proc ICIP*, vol. 3, p. 304–308, 1999.

ACKNOWLEDGEMENTS

We thank Malcolm Slaney: his comments, suggestions, and discussion guided us in clarifying this presentation.