

VIDEO DISSOLVE AND WIPE DETECTION VIA SPATIO-TEMPORAL IMAGES OF CHROMATIC HISTOGRAM DIFFERENCES

Mark S. Drew, Ze-Nian Li, and Xiang Zhong
School of Computing Science, Simon Fraser University,
Vancouver, B.C. Canada V5A 1S6
{mark,li,xzhong}@cs.sfu.ca

ABSTRACT

Gradual transitions represent a challenging problem for temporal segmentation of video. Here we present two new features for detecting these. Recently, Ngo et al. set out a method for edge detection in spatio-temporal images made out of the central column (or row, or diagonal) of a video. A wipe generates a diagonal edge in such an image. In this paper we make use of all available pixels to generate spatio-temporal images. For each column of the frame (using only the DC values from a video MPEG), we form a 2D histogram based on chromaticity, and then intersect that histogram with that of the previous frame (one or several frames earlier). The result is an image in which cuts and wipes appear as very strong edges, almost 1's in a background of zeroes. Dissolves require another approach; here we extend a color-distance based histogram metric due to Hafner et al. by applying the method to 2D Cb-Cr histograms and changing the definition so that the metric displays a *near-constant* value during a dissolve, and zero elsewhere. We show results on videos that include fast subject motion and camera movements.

1. INTRODUCTION

Video temporal segmentation is an important basic step for multimedia presentation and summarization strategies. Many interesting and effective schemes have been devised for finding abrupt cut transitions (see, e.g., [1, 2]). However, the problem of finding gradual scene transitions is more difficult (cf. [3]).

Recently, a spatio-temporal approach has been developed by Ngo et al. [4, 5] which uses one horizontal, one vertical, and one diagonal image row (or blurred set of a few rows) to construct three spatio-temporal images. In the case of a wipe, the spatio-temporal images display diagonal transitions that can be detected. These authors therefore perform edge enhancement on such edges and thus derive the beginning and ending frames for the wipe.

In the case of a dissolve, matters are not so simple: the authors consider the mean color in each of the three “slices” they employ, plus the variance, to try to characterize a dissolve. Nevertheless, it is straightforward to find counter-examples to their method, in that it is sensitive to camera movement and indeed movements within frames.

Here we extend the basic method of [4] in two fundamental ways. First, we make use of every slice available in a video, and not just one particular one for each direction. Thus we bring to bear all the information available. Second, we use a histogram-based differencing scheme for spatio-temporal image creation, rather than pixel values themselves. This makes the method much more robust to noise and artifacts that are not part of the gradual transition itself. Wipe detection is based on histogram intersections of coarse two-dimensional histograms for the chromaticity

of each column or row, or sliding window of a few rows, of the DC components of frames. We show that such a spatio-temporal chromaticity histogram metric provides far more accurate and robust performance for real videos.

For dissolve transitions, which are the most difficult to identify, we develop a new result based on histogram differences. Firstly, to maintain linearity we adopt the 2D Cb-Cr chromaticity space. In order for the method not to fail on very simple images, we go from simple histogram intersection to the more complex but perceptually more realistic histogram-difference metric developed by Hafner et al. [6]. Since each column or row of a frame consists of only a small number of DC components, we analytically re-write a variant of the Hafner metric explicitly in terms of pixel values, rather than histogram bin counts — this explicitly and exactly preserves temporal linearity if it indeed exists (whereas the original histogram definition does not). The result is that temporal differences for each column equal a constant times $(t_1 - t_2)^2$, with t_1 and t_2 the beginning and ending time we are examining for frame differences. This means that if $(t_1 - t_2) \equiv \Delta t$ is a constant, then the histogram-difference D^2 we produce is also a constant.

On the other hand, if t_1 is fixed, and $t_2 \equiv t$ varies, then we see a function D^2 which is a quadratic in time, and its square root is linear.

Moreover, again for the case of fixed $(t_1 - t_2) \equiv \Delta t$, the measure D^2 is composed of three parts, each of which is quadratic in time and hence has linear derivative. The three parts add to a constant D^2 . Since we examine each column difference separately, we have a feature that can be normalized to approximately 1 during a dissolve and 0 outside a dissolve, for each column separately, using column-based differences, and similarly for rows or diagonals.

Therefore we have overall a large number of indicators, including all rows, all columns, all diagonals, and both kinds of measure, to employ in determining dissolve transitions. Results are good, and better than previous methods.

2. WIPE AND CUT TRANSITIONS

In a wipe, one video is (linearly, usually) replaced over time by another video. Fig. 1(a) shows a typical transitional image. Taking pixels from the middle column and placing them sideways, we see a pixel-based spatio-temporal image, as in [4]. Fig. 1(b) shows such an image, for a video segment that includes wipes and cuts. While we might be able to roughly discern a diagonal edge at a wipe transition, we shall see that we can improve it greatly using histograms instead of pixel differences.

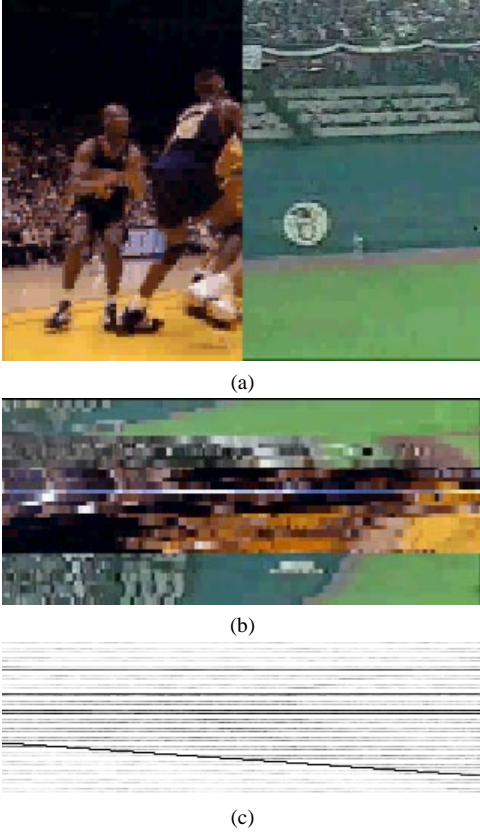


Fig. 1. (a): Wipe. (b): Pixel-based detection of a wipe and several cuts; time goes up the ordinate axis, and at any time each row is made out of the middle column in Fig. (a). (c): Spatial chromaticity histogram-based detection of wipe and cuts; again, time goes up the ordinate axis, but now each element of a row is composed of the histogram intersection for each of the columns in the image, separately.

Now, instead of using the pixels themselves, let us form a spatio-temporal image by first going over to two-dimensional chromaticity coordinates, $r = R/(R+G+B)$, $g = G/(R+G+B)$. This step effectively eliminates shading effects, as well as image intensity. Then let us form a two-dimensional chromaticity histogram, for each column (or row, or diagonal). Forming histogram differences using a histogram intersection (see [7]) is much more reliable than using pixel differences. Generally, if images are fairly slowly changing, in each video, then histogram intersection produces values that are nearly unity, if we compare a column at a later time with itself at a previous time. However, when the wipe reaches that column, the abrupt change yields a histogram intersection approximately equal to zero.

Fig. 1(c) shows how the edges appear using a chromaticity histogram intersection based difference scheme. At any time, each column has a scalar histogram intersection with the same column at the previous time. In the figure, time goes up the ordinate axis; and for each column in the image, the scalar for that column forms the image value. (A similar image is created if we use rows or diagonals instead of columns.) Therefore we are using all image information available, instead of just one column or row. Note that Fig. 1(c) is raw output, with no edge enhancement performed.

In Fig. 1(c) we see a much sharper edge for a wipe, which no edge enhancement technique could produce from Fig. 1(b). As

well, Fig. 1(c) shows the presence of cuts in the video, which appear as horizontal lines at the cut time. (In fact, the video shown has an abrupt flash of blue light, which shows up as two closely-placed cut lines.) The method does very well, even though the video contains a good deal of action, including a fast-moving ball and camera movements. In all videos we tried, including videos with rapid bird-wing movements, pans, zooms, etc., results were very similar to those shown above.

We could also use the histogram-intersection method on 2D histograms formed from Cb-Cr images. In fact, in videos we tested, results using Cb-Cr and chromaticity were very similar. Nevertheless, chromaticity should in principle produce better results because shading is removed from images.

3. DISSOLVE TRANSITIONS

Figs. 4 shows a dissolve. In a dissolve transition, one video blends smoothly into a second one. Without loss of generality, let us define the transition simply via

$$\mathbf{R} = \mathbf{A} + \alpha(t)(\mathbf{B} - \mathbf{A}), \quad (1)$$

where \mathbf{A} and \mathbf{B} are the 2-vectors for video A and video B, in Cb-Cr space. Here, $\alpha(t)$ is a transition function, which may be linear:

$$\alpha(t) = Kt, \quad \text{with } Kt_{max} \equiv 1. \quad (2)$$

The problem with using histogram intersections for temporal differencing of chrominance histograms of video columns is that this will fail in very simple cases. For if video A and video B are both uniformly-colored still images, then a Cb-Cr histogram consists of a single peak, which gradually (and usually linearly) moves from one color to the other. But usually this will produce zero histogram intersection.

Therefore we go over to a histogram-difference measure which obviates this problem, introduced by Hafner et al. [6].

In [6], a histogram-difference D^2 was defined as follows:

$$D^2 = \mathbf{z}^T \mathbf{A} \mathbf{z} \quad (3)$$

where

$$a_{ij} = (1 - d_{ij}/d_{max}) \quad (4)$$

and d_{ij} was defined as a three-dimensional color difference (using any metric desired). Vector \mathbf{z} is a histogram-difference vector (for vectorized histograms). For example, the histogram-difference vectors \mathbf{z} would be of length 256 if our chromaticity histograms were 16×16 .

Here, for simplicity and efficiency, firstly we go over to a two-dimensional Cb-Cr chrominance space. However, if we use a Euclidean or other color-difference metric d_{ij} , the above histogram difference measure will not be linear under a temporal transition (1) with linear $\alpha(t)$.

Therefore we set out a variant of (4) which does remain linear:

$$a_{ij} = (1 - d_{ij}^2/d_{max}^2) \quad (5)$$

Now suppose we use only DC components — then each frame column will consist of only one eighth of the number of rows in an image. Therefore it would be useful to go over to a method that duplicates eq. (3), but simply *analytically* uses the pixel values, rather than using a histogram of these.

To see how an analytic expression can be devised, suppose that somehow we had available an infinitely precise histogram; then the vector \mathbf{z} would consist of a 1 for each pixel in the first frame, and

a -1 for each pixel in the later frame. I.e., vectors z consist of entries as follows:

$$z = (1, 1, 1, \dots, -1, -1, -1, \dots) \quad (6)$$

where the 1's entries appear for pixels in the current column for the previous time, and the -1 entries appear for pixels in the current column for the current time.

Thus, vectors z in the expression (3) are very simple and the contributions from the 1 in a_{ij} in (5) in fact all cancel.

For the remaining terms, each 1 in z (corresponding to time t_1), at pixel i , say, must multiply another 1 at column j , for all i, j . As well, each -1 (corresponding to time t_2) multiplies other -1 's as well as the 1's. Therefore, we have

$$D^2 = (1/d_{max}^2) \{ \sum_i \sum_j \|R_{t_1}^i - R_{t_1}^j\|^2 - 2 \sum_i \sum_j \|R_{t_1}^i - R_{t_2}^j\|^2 + \sum_i \sum_j \|R_{t_2}^i - R_{t_2}^j\|^2 \} \quad (7)$$

where $R_{t_1}^i$ is the Cb-Cr 2-vector at time t_1 for the i th row in the current column, if we are differencing pixels in columns between time t_1 and time t_2 (and similarly for row or diagonal differencing).

Now let us substitute the dissolve definition (1), and use Euclidean distance in Cb-Cr space. Then (7) can be more simply expressed as

$$D^2 = (2/d_{max}^2) \sum_i \sum_j \{ R_{t_1}^i \cdot R_{t_1}^j - 2R_{t_1}^i \cdot R_{t_2}^j + R_{t_2}^i \cdot R_{t_2}^j \} \quad (8)$$

For a linear transition (2), some algebra and the use of symmetry in the expressions being summed leads to the following simple expression for D^2 :

$$D^2 = 2(1/d_{max}^2) K^2 (t_1 - t_2)^2 \sum_i \sum_j (B^i - A^i)^T (B^j - A^j) \quad (9)$$

Since the sum above is simply a constant (that depends on the pixel values in the column being used, for the beginning and ending video frames) we arrive at the conclusion that for constant $(t_1 - t_2) = \Delta t$, the difference D^2 is a *constant* over the transition (under the assumption that the video does not change greatly during the dissolve). Before and after the transition, D^2 is approximately zero.

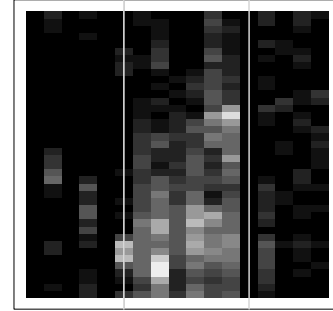
If t_1 is instead taken to be the beginning time of the transition, then D^2 is a quadratic function of time time $t_2 \equiv t$. Its square root is linear in time.

As well, for constant $(t_1 - t_2) = \Delta t$ again, each term in (7) is separately quadratic in time with linear derivative, and this can provide additional evidence for a dissolve transition.

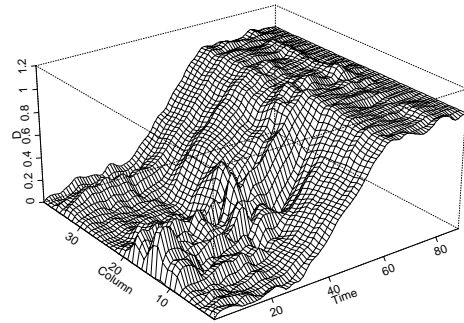
Figs. 2 shows the D^2 result for a video of a hummingbird, dissolving into a windy flower-garden scene. In Fig. 2(a), we have divided D^2 by the value of expression (7) for t_1 and t_2 equal to the beginning and end of the sequence — the feature has a strong jump at each dissolve transition boundary. We difference frames that are 5 frames apart. The value by which we divide is different in each column. By dividing, we arrive at a feature that approximately equals 1 during the dissolve and 0 outside it, in each column. From the Figure, we see that this algorithm has performed fairly well in characterizing the dissolve.

In Fig. 2(b), we show the linear square root of quadratic D^2 that results when t_1 in eq.(7) is taken to be the first frame, rather than the previous frame. We difference frames that are 1 frame apart. These kinds of results are typical for videos we examined.

Here we clearly see the linear slope that the square root of eq.(9) represents when $(t_1 - t_2)$ is not a constant, as in Fig. 2(a), but instead is proportional to the difference in time between the current frame and the first frame. We determined in general that the method of comparing to the first frame found for the dissolve was more robust than the method of using an incremental comparison.



(a)



(b)

Fig. 2. Dissolve for video of a hummingbird and garden. (a): Value of D^2 , differencing frame to previous frame. 5 frame interval. Vertical lines show actual transition boundaries. Function is approximately constant over the transition and zero elsewhere. (b): Square root of D^2 , differencing frame to initial frame. 1 frame interval; function is approximately linear.

In comparison to previous work, we found that the average R,G, and B values for each column, advocated by [4] for a single column, are linear in principle but are in practice far more susceptible to the effects of noise and motion in the video. The new feature presented here is more stable because it is essentially histogram-based. If we were to consider only mean color, then a range of colors over a column that averaged to a gray, blending with another such video, would not show a linear change, whereas our new method would. For example, Fig. 3 shows the average red channel, for each column in the frame, over a dissolve that goes from frame 30 to frame 60. Here we have used DC values with

a 1-frame interval, and divided by the mean color in each column separately at the end of the sequence. We tried clamping the initial frame to zero, but the very large range of deviations from a linear change swamps attempts to make each transition into a line from 0 to 1.

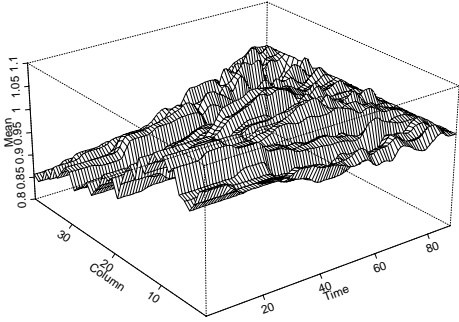


Fig. 3. Average color for dissolve: divide by last frame to make each column end at same average.



Fig. 4. Dissolve.

We could also use the method we have devised for dissolves, based on eq.(7), to operate on wipes and cuts, as well. In this case, one again sees a sharp diagonal line, for a wipe, with very abrupt values that are column- (or row-, or diagonal-) dependent. However, the method based on histogram intersections is more efficient than that based on eq.(7).

Nevertheless, eq.(8) can in fact be made more efficient for a vector machine: suppose we collect all $N \times 2$ vectors $\mathbf{R}_{t_1}^i$, $i = 1..N$, for the current column (or averaged sliding window of columns) into a $2 \times N$ array ρ :

$$\rho_1 = \begin{pmatrix} Cb^1 & Cb^2 & Cb^3 & \dots & Cb^N \\ Cr^1 & Cr^2 & Cr^3 & \dots & Cr^N \end{pmatrix} \quad (10)$$

where ρ_1 pertains to time t_1 . Then the feature eq.(8) can be simply written vectorwise:

$$\text{sum}(\rho_1^T \rho_1 - 2\rho_1^T \rho_2 + \rho_2^T \rho_2) \quad (11)$$

4. CONCLUSIONS

We have provided two new measures for detecting cuts, wipes, and dissolves. The new measures are simple in that they involve only simple summations — e.g., one forms the sums in (8), or over column histograms. The use of multiple rows and columns provides a large number of descriptors for gradual transition detection.

For wipes and cuts, we found the use of histogram intersection of chromaticity histograms most effective. Histogram intersection is a very fast method. Dissolves present more of a challenge, and we have set out a method here for a simple test for dissolves: if the feature in eq.(8) displays constant behavior in each column (or row, etc.), then a dissolve is present. Further substantiation can be made using the linear behavior of the square root of the feature if we compute frame-to-beginning-frame differences, rather than frame-to-frame. The difference from the initial frame of the dissolve is a linear ramp. The frame-to-beginning-frame measure is more stable than the frame-to-frame difference; the latter is most useful for detecting the boundaries of a dissolve, which show up as strong jumps.

In a sense, we have shifted the computational burden away from edge detection, and onto feature computation. However, the feature computed is quite reliable, and very little edge-detection is necessary to find the gradual transition boundaries.

These features perform best when the assumption that each video in a dissolve does not change much over the course of the dissolve is not violated. This would not be the case if there was substantial motion, a circumstance made more likely if the dissolve is a long one. In general, we found the wipe-detection scheme to be very robust, and the dissolve-detection feature to work best for (1) larger averaging implied by using DC values rather than pixels themselves, and (2) longer time-intervals between comparisons, say 5 frames, rather than 1.

5. REFERENCES

- [1] M. S. Drew, J. Wei, and Z.N. Li, "Illumination-invariant image retrieval and video segmentation," *Pattern Recognition*, vol. 32, pp. 1369–1388, 1999.
- [2] A.M. Ferman and A.M. Tekalp, "Efficient filtering and clustering methods for temporal video segmentation and visual summarization," *J. Vis. Commun. & Image Rep.*, vol. 9, pp. 336–351, 1998, Sp. Issue on Multimedia Storage and Archiving Systems.
- [3] H.J. Zhang, A. Kankanhalli, and S.S. Smoliar, "Automatic partitioning of full-motion video," *Multimedia Systems*, vol. 1, pp. 10–28, 1993.
- [4] C.W. Ngo, T.C. Pong, and R.T. Chin, "Detection of gradual transitions through temporal slice analysis," in *CVPR99*, 1999, pp. I36–41.
- [5] C.W. Ngo, T.C. Pong, and R.T. Chin, "A robust wipe detection algorithm," in *ACCV2000*, 2000, pp. 246–251.
- [6] J. Hafner, H.S. Sawhney, W. Equitz, M. Flickner, and W. Niblack, "Efficient color histogram indexing for quadratic form distance functions," *IEEE Trans. Patt. Anal. and Mach. Intell.*, vol. 17, pp. 729–736, 1995.
- [7] M.J. Swain and D.H. Ballard, "Color indexing," *Int. J. Comput. Vision*, vol. 7, no. 1, pp. 11–32, 1991.