Comparison of automatic video segmentation algorithms

Apostolos Dailianas
Department of Computer Science, Columbia University *
Robert B. Allen
Bellcore, Morristown NJ
Paul England
Bellcore, Morristown NJ

ABSTRACT

While several methods of automatic video segmentation for the identification of shot transitions have been proposed, they have not been systematically compared. We examine several segmentation techniques across different types of videos. Each of these techniques defines a measure of dissimilarity between successive frames which is then compared to a threshold. Dissimilarity values exceeding the threshold identify shot transitions. The techniques are compared in terms of the percentage of correct and false identifications for various thresholds, their sensitivity to the threshold value, their performance across different types of video, their ability to identify complicated transition effects, and their requirements for computational resources. Finally, the definition of a priori set of values for the threshold parameter is also examined. Most techniques can identify over 90% of the real shot transitions but have a high percentage of false positives. Reducing the false positives was a major challenge, and we introduced a local filtering technique that was fairly effective.

1 INTRODUCTION

To extract useful information from a digital video source, the information contained in it has to be indexed and annotated. Examples of services that can be provided after indexing include video databases, video browser capable of presenting information in an organized manner[3,9,12] and others. Typically the annotation of the information in the video has to be associated with points in the video where a change of subject occurs. Usually, but not always, the change of subject coincides with shot transitions, where a *shot* is defined as the sequence of successive frames from the moment a camera starts recording until it stops. During the shot duration, both the camera and the objects in the image can move, but the sequence of frames is part of the same shot. Therefore, The first step in indexing digital video is the identification of shot transitions. The manual identification of shot transitions is a very time consuming process, and is therefore expensive and impractical.

A variety of techniques for automatic video segmentation has been developed, but these have not been systematically compared. This paper compares different methods for identifying shot transitions concentrating on methods that perform minimal computations on the values of pixels in consecutive frames within an image sequence. Such methods are much faster than those involving image analysis and object recognition, while still

*This work was completed during the summer of 1995 when the first author was visiting Bellcore. Correspondence can be addressed to apostolo@cs.columbia.edu, rba@bellcore.com, or england@bellcore.com.

providing good performance in terms of identifying shot transitions. The comparison of the different methods is performed in terms of their ability to identify shot transitions and their sensitivity to a threshold parameter used for decision making.

One difficulty in automatic video segmentation stems from the variety of techniques used to create transitions between shots. The simplest of these is the *cut* in which frames of one shot are immediately followed by frames of the next shot. A *fade* to a color (usually black) is a gradual transition of the frames of a shot that fades out to that color followed by the gradual transition of the frames of the shot that fades in from that color. In a *dissolve* the two effects (fade out and fade in) overlap.

One class of segmentation methods examined, is based on the comparison of histograms of color intensities from consecutive frames. Several variations of this technique, like the pure histogram difference, the squared difference, the difference after equalization, the intersection of histograms, and others are studied. The second class of methods is based on the extraction of moment invariants which possess some useful properties including invariance to scale change, rotation, and translation. The third class of methods is the classification of frames of an image sequence based on mathematical models; one of the models examined is based on the analysis of the statistics governing the change of value of a given pixel in consecutive frames. Finally, a method that performs edge recognition is also presented and analyzed. Because almost all of the methods have a threshold parameter which serves to distinguish between shot changes and differences due to the evolution of the image sequence within a shot, an important characteristic is the robustness of the method to the value of this threshold. A plot of the correct and false detections versus the value of the threshold indicates the region of threshold values for which each of the methods performs acceptably.

Another important comparison, related to the ability of real-time or close to real-time implementations of the different methods, is the number of computations performed to compare two consecutive frames. A comparison related to both the robustness to the threshold parameter and the computational burden of the method is the sensitivity to spatial under-sampling which may greatly reduce the time needed by the methods and still lead to good results.

2 SEGMENTATION METHODS

All the methods in this section, examine two frames f and f' and define a measure of dissimilarity $d(f, f')$ whose value can be used to identify shot transitions.

2.1 Absolute Frame-Difference segmentation

The simplest measure of the difference between two consecutive frames is the absolute difference of the sum of the intensities of all pixels in the frame. The grey-level intensity of a pixel is defined as $0.299r + 0.587g + 0.114b$ for a color image, where r, g and b are the intensities for the three basic colors, each represented by a one-byte number.

2.2 Histogram-based segmentation

Among the simplest, most effective, and most commonly used methods, the histogram-based method and its many variations[3,7–9,11] rely on the following basic idea: The number of values each pixel can have is discretized and a histogram is created for a frame by counting the number of times each of the discrete values appears in the

frame. Then, this histogram is compared with the histogram of the next frame. Histograms of frames within the same shot should be very similar to each other, even in the case of camera or object motion, because the method is relatively insensitive to the position of objects within the frame.

Let $H(f,k)$ be the value of the histogram for frame f and for the discrete value of intensity k. The value of k is in the range $[0,N]$, where N is the number of discrete values a pixel can have. There are usually three histograms for each frame, one for each color. Several variations of the histogram method are described in the following subsections.

2.2.1 Simple histogram difference

The metric of the dissimilarity between frames f and f' is given by: $d(f,f') = \sum_{j=0}^{N} |H(f,j) - H(f',j)|$.

2.2.2 Weighted histogram difference

In a frame there might be some dominant color which should be given a greater weight in the comparison between two frames. Thus, we developed a formula for the weighted histogram difference which is defined as:

$$d(f,f') = \frac{r}{s} \cdot d(f,f')^{(red)} + \frac{g}{s} \cdot d(f,f')^{(green)} + \frac{b}{s} \cdot d(f,f')^{(blue)}$$

where r, g and b are the luminance for the red, green, and blue components of the picture respectively and s is defined as $(r+g+b)/3$.

2.2.3 Histogram difference after equalization

The goal of histogram equalization[1,5] is to produce a uniform histogram $H^e(f,j)$ for the output frame. Then,

$$d(f,f') = \sum_{j=0}^{N} |H^e(f,j) - H^e(f',j)|.$$

where the equalized histogram $H^e(f,j)$ is obtained in the same way as $H(f,j)$ after transforming the value v of a pixel to v_{eq} through the following procedure:

$$v_{eq} = Int[\frac{w - w_{min}}{1 - w_{min}} \cdot (L-1) + 0.5]$$

where N is the number of levels of the pixel value, w is given through:

$$w = \frac{1}{\sum_{j=0}^{N-1} H(f,j)} \sum_{j=0}^{v} H(f,j)$$

and w_{min} is the smallest positive value of w.

2.2.4 Intersection of histograms

The intersection of two histograms,[10,11] which also serves as the metric of the similarity $s(f, f')$ between two frames, is defined as:

$$IS^{(color)}(f, f') = s(f, f') = \sum_{j=0}^{N} \min(H(f, j), H(f', j))$$

When taking the intersection of two identical frames, the above value is maximum and equal to the number of pixels in the frame, whereas in dissimilar frames this value is generally much lower. Uniform treatment of the techniques requires the definition of a dissimilarity metric. Letting M represent the maximum value of similarity between any two consecutive frames in the video, $d(f, f')$ can be defined as: $d(f, f') = M - s(f, f')$.

2.2.5 Squared histogram difference

Compared to the pure histogram difference, this method[7] tries to amplify the difference of two frames using the following formula:

$$d(f, f') = \sum_{j=0}^{N} \frac{(H(f, j) - H(f', j))^2}{H(f, j)}.$$

The division by $H(f, j)$ serves as a normalization factor for the significance of the square of the difference on the nominator. Our preliminary tests showed that a more effective variation of this formula is:

$$d(f, f') = \sum_{j=0}^{N} \frac{(H(f, j) - H(f', j))^2}{max(H(f, j), H(f', j))}.$$

2.3 Segmentation based on moment invariants

Moment invariants have properties such as invariance to scale change, rotation, and transition, which make them good candidates as a representation of the frame. This technique was introduced by[2] who report good results when it is used in combination with the intersection of histograms method. The moment of an image $f(x, y)$ is defined as:

$$m_{pq} = \sum_{x} \sum_{y} x^p \cdot y^q \cdot f(x, y).$$

Moment invariants are derived from normalized central moments defined as:

$$n_{pq} = \frac{1}{m_{00}^{\gamma}} \cdot \sum_{x} \sum_{y} (x - \bar{x})^p \cdot (y - \bar{y})^q \cdot f(x, y)$$

where $\gamma = 1 + (p + q)/2$, $\bar{x} = m_{10}/m_{00}$ and $\bar{y} = m_{01}/m_{00}$.

In this study, we applied the first three moment invariants. These are defined as:

$$\phi_1 = n_{20} + n_{02}$$

$$\phi_2 = (n_{20} - n_{02})^2 + 4n_{11}^2$$

$$\phi_3 = (n_{30} - 3n_{12})^2 + (3n_{21} - n_{03})^2$$

The Euclidean distance can be used as the metric of the difference between two frames: $d(f, f') = |\vec{\sigma}_f - \vec{\sigma}_{f'}|^2$, where $\vec{\sigma} = \{\phi_1, \phi_2, \phi_3\}$.

2.4 Segmentation based on the range of pixel-value changes

The method proposed in[1] models the difference between the values of a pixel in two successive frames as the combination of three factors: The first is a small amplitude additive zero-centered gaussian noise, modeling camera, tape, and digitizer noise. The second is the change of the pixel value resulting from object or camera motion, change of focus and lightning at a given time in a given shot. The third is the change caused by a cut, wipe, dissolve, or fade to/from black or white. According to the analytical models for each of the three factors above, cuts can be found by looking at the number of pixels whose difference of value in two consecutive frames falls in the range [128,255], whereas dissolves and fades to/from color can be identified by the number of pixels whose change of value between consecutive frames falls in the range [7,40] for 8-bit coded grey-level images. Finally, wipes are also identified by the number of pixels with value changes in the range [128,255], but their identification is not reliable. Histogram equalization (Section 2.2.3) is performed for cut and wipe detection but not for fade detection. The method is not intended for static detection of shot transitions by comparing two frames. Instead, it incorporates temporal filtering of the values in the above mentioned ranges, over a sequence of consecutive frames.

2.5 Segmentation based on edge detection

The method proposed in[6] is based on the observation that during a shot transition new intensity edges appear far from the locations of old edges, and old edges disappear far from the location of new edges. Therefore, shot transitions can be identified by comparing the edges in two consecutive frames. The method also uses a registration technique to compute an overall motion between frames, which is taken into account in the computation of the percentage of new edges far away from the old ones. Letting ρ_{in} denote the percentage of edge pixels in frame f which are more than a fixed distance r from the closest edge pixel in frame f', and ρ_{out} the percentage of edge pixels in frame f' which are farther than r away from the closest edge pixel in frame, the metric used to measure dissimilarity between two frames[6] is $d(f, f') = max(\rho_{in}, \rho_{out})$.

3 COMPARISON OF METHODS

3.1 Procedure

3.1.1 Videos

Four test videos were used; these included 2 news broadcasts a training video, and a feature movie. These were originally on NTSC tape and were digitized in Intel Indeo 3.2 format. The news videos were the ABC Evening News for May 9, 1995 and the CBS-affiliate local news for August 2, 1995. These also included commercials, which were particularly difficult to segment. The training video was prepared by Bellcore Training and Education (TEC) and was about ATM. The feature movie was *Out of Africa*.

3.1.2 Segmentation by human observers

It is necessary to have some independent standard by which to compare the segmentation algorithms. Thus, we had a human observer determine where the segment boundaries were. For the ABC news broadcast, the most complex of the videos, a second observer also performed the segmentations. The human-observer segmentation is subjective, because the shot transitions are often ambiguous. For example, for the news broadcast or the training

Video	digitization rate (frames per sec)	duration (min)	cuts	fades, dissolves, and wipes
ABC Evening News	15	28.5	353	160
CBS local news	12	30	366	115
ATM training	12	60	147	7
Out of Africa	12	31	275	9

Table 1: Characteristics of the videos.

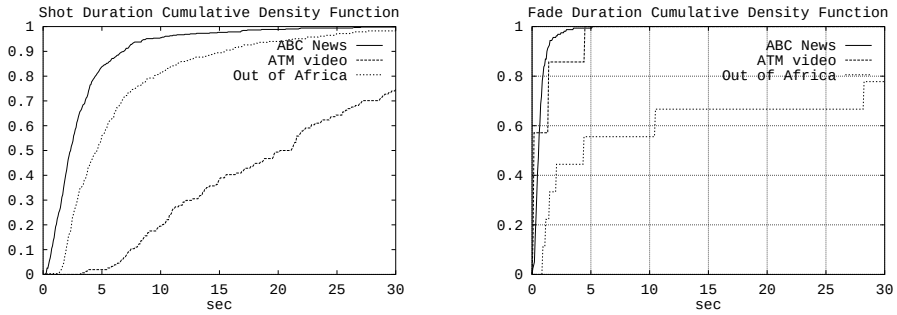


Figure 1: Shot and fade durations.

videos, inset of material into the frame, changes only part of the picture, but that part changes extensively. Such effects are identified as shot transitions by some observers but not by others. For the ABC news video, with over 500 shot transitions, there were only about six cases in which the two observers differed.

As shown in Figure 1, the videos have very different characteristics. The news videos were by far the hardest to segment. The most difficult parts of the news broadcast to segment were the commercials. Some of the commercials have very short shots dissolved with one another and sometimes have more than two shots overlapping with each other in a dissolve. The training video was easy to segment while the feature movie was mostly straightforward, but some fades were extremely long. The output of the human segmentation is the identification of shot transitions and their classification as either cuts or other effects (e.g., fades, dissolves, and wipes) are presented in Table 1. Section 3.3, the segments identified by the human observer are compared to the segments identified by the automatic methods.

3.1.3 Setting thresholds for automatic segmentation

For all of the methods except those presented in Sections 2.4 and 2.5, we adopted the following automatic shot-transition identification method. A threshold is specified and the values of the similarity metric are compared against it. Whenever the values cross the threshold from below to above, a shot transition is identified. The methods in Sections 2.4 and 2.5 use prespecified parameters to determine whether a shot transition has occurred.

3.2 Filtering algorithm

Initial tests of the segmentation methods showed that many of the shot transitions triggered spurious shot-transition identifications. One of the most common cases of spurious shot transition was caused by the threshold being crossed multiple times for a single transition due to variations of the value of the metric. In Section 3.6, other common reasons for failure are examined. Although little can be done to identify shot transitions which were not found due to producing small values for the metric of dissimilarity between consecutive frames, a lot can be done to reduce false shot-transition identifications.

We designed a simple filter that processes the sequence of values of the metric of dissimilarity for the different methods and produces a new sequence with the following procedure. The value is compared with the k previous values and the k following values. If any of these $2k$ values is greater than the current value, replace the current value with the last local minimum detected in the sequence of metric values.

Two conflicting requirements govern the choice of k. Shot transitions that take place over a sequence of frames (e.g., fades, dissolves, wipes) can cause big oscillations in the sequence of values of the metric of dissimilarity, resulting in several crossings of the threshold for the same transition effect. The value of $2k$ should ideally be larger than the longest of those transition effects. On the other hand, there exist shots whose duration is shorter than the longest transition effect. The ability to identify them, requires that $2k$ is smaller than the duration of the shortest shot. A reasonable value of k can be obtained from examination of Figure 1 where the cumulative histogram of the duration of shots and the duration of fades is shown for the three videos. Based on this figure and the fact that the ABC News video was digitized at 15 frames per second, k has been set to 5. Notice that the videos have very different values for the shot durations. The choice of k for the filter is selected according to the statistics of the news video, because a larger k suggested by the statistics of the ATM training video would filter out real shot transitions that are close to each other.

An alternative to the filter proposed in this section, would be a moving average window, with the window size specified with reference to Figure 1. The moving average technique can shift the peak values of the metric, introducing some inaccuracy in the detection of the shot transition instant. Furthermore, the problem of multiple crossings of the threshold for the same transition is alleviated but not completely gone, because for small window sizes as those suggested by Figure 1, the variations in the value of the metric may still exist.

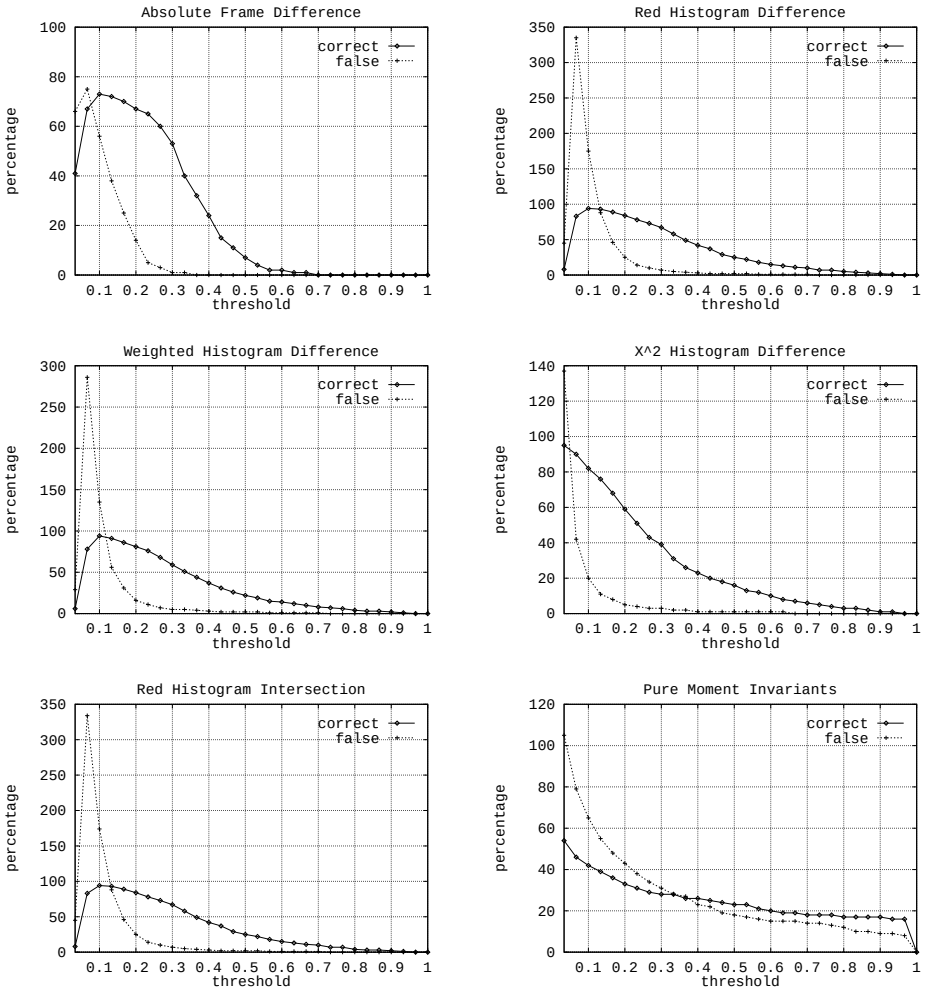
Another promising technique trying to overcome the false identifications related to a global threshold, is *local thresholding*. In local thresholding, the value of the threshold changes over time according to the characteristics of the metric within an observation window. The problem of multiple crossings of the threshold can easily be overcome by setting the local threshold to an appropriate value beyond the local averages of the metric values. An example of local thresholding can be found in[1] where a metric value identifies a cut when it is larger than approximately twice the average value of the metric in a window of previous values that extends at most up to the previous cut (for exact values and formulas see[1]). In the same paper, a local threshold technique is also applied to the identification of fades.

Both global and local thresholding can easily be coupled with simple filters to suppress the identification of spurious transitions. An example can be found in,[1] where a metric value is considered as a candidate cut-transition point only if it is much bigger than the maximum of the neighboring values.

3.3 Comparison of automatic methods

This section compares the performance of the video segmentation techniques for the ABC news video which was chosen because it contains a variety of shot-transition effects such as cuts, fades, dissolves, and wipes. It also contains advertisements which are generally hard to automatically segment due to the short duration of the shots

which are dissolved with each other, and also because there is often fast object or camera motion. Finally, some parts of the video come from old, black-and-white films which are very noisy. The filter described in Section 3.2 has been applied to all of the methods where the segmentation is based on the use of a global threshold.



Method	% Correct identifications	% False identifications
Range of pixel-value changes	95	68
Edge detection	92	59

Figure 2: Correct and false identifications for ABC news video.

Figure 2 shows the percentage of correctly identified shot transitions and the percentage of the false identifications for each of the methods. The horizontal axis is the value of the threshold used as the parameter to decide where shot transitions occur. It ranges from 0 to 1, with 1 corresponding to the highest value found in the metric. The vertical axis shows the percentage of correctly or falsely identified shot transitions relative to the real number of shot transitions which corresponds to 100%. While separate histograms could have been computed for each color, we focus on red histograms because they are representative of, or perhaps slightly better than, the other colors. Notice the very high percentage of false identifications for low thresholds, even after the application of the filtering algorithm.

Method	% Correct identifications	% False identifications
Absolute Frame Difference	73	56
Red Histogram Difference	94	175
Weighted Histogram Difference	94	135
X^2 Red Histogram Difference	95	137
Red Histogram Intersection	94	174
Pure Moment Invariants	54	105
Range of Pixel-Value Changes	95	68
Edge Detection	92	59

Table 2: Maximum % of correct identifications and corresponding % of false identifications for ABC News video.

Table 2 compares the different methods in terms of the percentage of correct and false identifications for the threshold that maximizes the percentage of correctly identified shot transitions. It can be seen that several methods identify correctly 94% to 95% of the real transitions. The corresponding high percentages of incorrect identifications are not very discouraging since incorrect identifications can easily be filtered out by the human who annotates the important shot transitions, and moreover this percentage reduces much faster than the percentage of correct identifications as the value of the threshold is increased (for example, for the Red Histogram Difference method in Figure 2, the percentages of correct and false identifications for the first point to the right of the point where the percentage of correct identifications is maximum are 93% and 88% compared to 94% and 175% for its adjacent point). The 68% false identifications for the Range of Pixel-Value Changes method appears unusually low and, indeed, we did not find such good performance with that algorithm on the other videos, even though the percentage of false identifications for this method was always among the lowest. The results for the Edge Detection method have been obtained using the code we were granted by the authors of.[6] The code was still under development, so better results may be expected with newer versions.

Figure 3 discriminates the correctly identified shot transitions between cuts and other shot transition effects (including fades, dissolves, and wipes). Although according to most of the metrics, fades, dissolves, and wipes should be much harder to detect because the changes happen gradually, it turns out that most of the methods do comparably well for cut and fade detection with the exception of the absolute frame difference method which performs very well for cuts but very poorly for other effects. Notice also that the threshold that maximizes the correctly identified cuts is different from the one for fades, dissolves, and wipes, with the later being lower as expected. Thus, in practice we believe that separate thresholds should be employed for the two types of effects.

The evaluation of the performance of the segmentation algorithms depends on several factors. The first of these is how the segmentation is going to be used. An application involving annotation by a human observer would care more for a high percentage of correct identifications than for a low percentage of incorrect identifications. When a human does not intervene, a low percentage of incorrect identifications becomes increasingly important. A second factor for evaluating methods that involve a threshold parameter, is related to the a priori specification of the threshold value, according to the results obtained in an experimental stage where the best threshold is specified for several videos. Therefore, as should be clear by looking at Figure 2, the *sensitivity* to the threshold parameter is an important criterion. A final factor is the type of video to be segmented also influences the choice of algorithm.

Method	ABC News	ATM Training	*Out of Africa*	Average optimal threshold
Absolute Frame Difference	0.10	$0.10 - 0.17$	0.20	0.15
Red Histogram Difference	0.10	$0.17 - 0.23$	$0.10 - 0.17$	0.14
Weighted Histogram Difference	0.10	0.17	$0.13 - 0.20$	0.15
X^2 Red Histogram Difference	0.03	0.07	$0.03 - 0.07$	0.05
Red Histogram Intersection	0.10	$0.13 - 0.20$	$0.10 - 0.17$	0.14
Pure Moment Invariants	0.03	0.03	0.03	0.03
Range of Pixel-Value Changes	independent	independent	independent	independent
Edge Detection	independent	independent	independent	independent

Table 3: Optimal thresholds for three videos and the average optimal threshold.

3.4 Variation of segmentation performance for different videos

In Section 3.3, the methods were compared for a given video. This section compares the performance of each segmentation method for different video material. Figure 4 compares the performance of each method for three of the videos. The horizontal axis is the threshold value relative to the maximum value of a method for a particular video. The vertical axis is the percentage of correctly identified shot transitions.

Figure 4 clearly shows that the ABC news video is the hardest to segment. The percentage of correct identifications is not only lower than in the rest of the videos, but it is also very sensitive to the threshold parameter implying that the threshold parameter does not transfer well across different types of video. For the difference in performance, consider the absolute frame difference method which performs very well for the ATM and the *Out of Africa* videos but not for the news broadcast. The difference in the threshold that maximizes the percentage of correct identifications and also the different sensitivity to the value of this threshold, can be seen in the Weighted Histogram Difference method. The methods that depend on the Range of Pixel-Value Changes and on Edge Detection seem to perform uniformly well across the different types of videos.

3.5 Automatically setting threshold parameters

Section 3.4 clearly shows that the threshold where the percentage of correct identifications is maximized (optimal threshold) is video dependent. The values for the optimal threshold are depicted in Table 3 for each of the methods. The fourth column in Table 3 is the average of the first three columns and tests the value of previously determined threshold for segmenting a new video. In order to appreciate the suitability of these values as a priori set thresholds and test whether the thresholds would be more stable within a type of video, we applied the automatic segmentation methods on the CBS news video. The results presented in Table 4 show that the average optimal thresholds from the fourth column of Table 3 give results which are close to optimal for the CBS news video.

3.6 Reasons for misses and false positives

Misses of shot transitions and false positives were examined in order to get a better understanding of the reasons why the methods fail to produce the desired results. The most common causes of misses are the following:

Method	CBS news optimal threshold	% Correct identifications for optimal threshold	% Correct identifications for average optimal thres.	% Correct identifications for ABC news optimal thres.
Absolute Frame Difference	0.10	81	79	81
Red Histogram Difference	$0.13 - 0.17$	92	92	90
Weighted Histogram Difference	0.13	92	91	89
X^2 Red Histogram Difference	0.07	92	91	90
Red Histogram Intersection	$0.13 - 0.17$	92	92	90
Pure Moment Invariants	0.03	60	60	60

Table 4: Test of various threshold values for CBS news video.

- Wipes are quite difficult to identify. None of the segmentation methods is able to distinguish clearly between a wipe and a sequence of images where objects are moving around quickly.
- The value of the specific metric is sometimes lower for a shot transition than for a non-shot transition. This is particularly true for cuts where the frames of the two shots are very close according to the metric. It also occurs for slow fades.

The most common causes of false positives are the following:

- High values of the dissimilarity metric for rapidly moving images.
- Sudden variations of the luminance of the image, either due to lightning effects (e.g., lightning) or to poor video quality.
- Fade out and then fade in (e.g., between advertisements) may be identified as two shot transitions although there is really only one shot transition.
- Variations in the value of the metric of dissimilarity between consecutive frames in the case of fades, dissolves, and wipes, cause the upward crossing of the threshold more than once for a single shot transition.

Local threshold techniques can help to reduce misses by small values for the metric of dissimilarity between consecutive frames. False shot-transition identifications can be greatly reduced with the filtering technique introduced in Section 3.2, or averaging of the values of the metric within a sliding window.

4 COMPUTATIONAL RESOURCES

When real-time segmentation of videos is required, an estimate of the number of operations to evaluate the measure of dissimilarity between two frames is required. Table 5 presents a rough estimate of the required computations per frame, under the assumption that addition, subtraction and multiplication require time equivalent to one operation, whereas divisions take approximately four times more. Implementation issues such as assignment of variables to registers, use of pointers or arrays, memory access time and others, are ignored. The variable N is the number of levels (bins) of the pixel value, and P is the number of pixels per frame. In both the Absolute Frame Difference and the Range of Pixel-Value Changes methods, we assume that grey-scale pixel values are provided by the video encoding. No estimate for the requirements of the Edge Detection algorithm was computed. The

code provided to us by the authors of[6] runs at over 4.5 frames per second on a *sparc10* machine, but it is still under development, and faster versions should be expected.

Method	Number of operations
Absolute Frame Difference	P
Red Histogram Difference	$P + 2N$
Weighted Histogram Difference	$6P + 3N$
X^2 Red Histogram Difference	$P + 7N$
Red Histogram Intersection	$P + N$
Pure Moment Invariants	$38P$
Range of Pixel-Value Change	$3P + 8N$

Table 5: Computational requirements for several methods.

5 DISCUSSION AND OPEN PROBLEMS

Because all of the methods studied here have high false-identification rates, they should be thought of as providing suggestions to human observers and not as an ultimate standard of performance. For instance, they could provide input to an interactive interface for specifying the structure of a video such as described by.[3] The high percentage of correct identifications of these methods, implies that for applications involving human interaction, computationally more expensive methods such as those involving object recognition could be avoided. A research area closely related to the adequacy of the methods described in this paper, is the development of filtering techniques like the one presented in Section 3.2 and the development of local thresholding techniques.

The choice of the best segmentation method is not straightforward. The time requirements, the sensitivity to the threshold parameter, the percentage of correct and false identifications and the type of videos to be segmented should by taken into account. Also for time-critical applications the performance of each method when spatial (within a frame) or temporal (across consecutive frames) undersampling is used, should be studied. It should be noted that some methods[1,2] are robust and can even improve by working at low resolution. Finally, the type of application segmentation is needed for, should also be considered. For example, an application for retrieval of shots based on video similarity could take advantage of the histogram information that can be stored during preprocessing of the video for a representative frame of each shot, to perform very fast retrieval of related shots (e.g., for the Red Histogram Difference method, each such comparison would require approximately $2N$ computations). Saving method-dependent information during preprocessing for the other methods can also help but they are still going to be much slower than histogram-based methods. Also, methods like the Range of Pixel-Value Change that was not intended to be applied for static detection between two frames, would be slow to use in such an application, since its decision is based on the temporal characteristics of some statistical measure over a sequence of consecutive frames.

The techniques presented here will be mostly useful for archival material. Given the problem of increasingly complex video production which includes more fades, wipes overlays and other digital shot-transition effects, and the ease of including an electronic segmentation mark on newly produced material, we suggest that shot transition information be broadcast along with the video.

There are still problems that the methods described above fail to address. One problem is how to distinguish between a fast change of the image in the same shot caused by movement of the camera (e.g., filming from within a moving car) and a cut or dissolve. Methods based on only simple statistical characteristics of the frame sequence might not be sufficient in this case and some more involved processing like object recognition might be necessary.

Another important problem is the identification of semantic transitions not associated with shot transitions. This situation is very common in news programs, where in switching between topics, only a small portion of the frame changes (usually a square in the background of the anchor) and this would normally not result in a transition identification.

6 ACKNOWLEDGMENTS

We thank Sheila Borack for segmentation of the videos and members of video server group for their assistance in developing the hardware. We also thank the authors of several of the segmentation techniques for email correspondence clarifying details of the use of those techniques. Finally, we would like to thank Kevin Mai, Justin Miller, and Ramin Zabih for providing their code for the Edge Detection method and Philippe Aigrain for his useful comments.

7 REFERENCES

[1] P. Aigrain and P. Joly, "The automatic real-time analysis of film editing and transition effects and its applications," *Computer & Graphics*, Vol. 18, No. 1, pp. 93-103, 1994.

[2] F. Arman, R. Depommier, A. Hsu, and M-Y. Chiu, "Content-based browsing of video sequences," *Proceedings of ACM Multimedia* (Nov. 1994) San Francisco, pp. 97-103.

[3] P. England, R.B. Allen, A. Dailianas, M. Sullivan, M. Bianchi, and A. Heybey, "The video library toolkit - A system for indexing and browsing digital video libraries." *Proceedings SPIE Photonics West'96: Electronic Imaging Science and Technology '96: Storage and Retrieval for Image and Video Databases IV*, San Jose (Jan. 1996).

[4] A. Hampapur, R. Jain, and T. Weymouth, "Digital video segmentation," *Proceedings ACM Multimedia* (Nov. 1994), San Francisco, pp. 357-364.

[5] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, 1989.

[6] K. Mai, J. Miller, and R. Zabih, "A robust method for detecting cuts and dissolves in video sequences," *Proceedings of ACM Multimedia* (Nov. 1995), San Francisco, to appear.

[7] A. Nagasaka and Y. Tanaka, "Automatic video indexing and full-video search for object appearances," *Visual Database Systems II*, Elsevier Science Publisher (North-Holland), 1992, pp. 113-127.

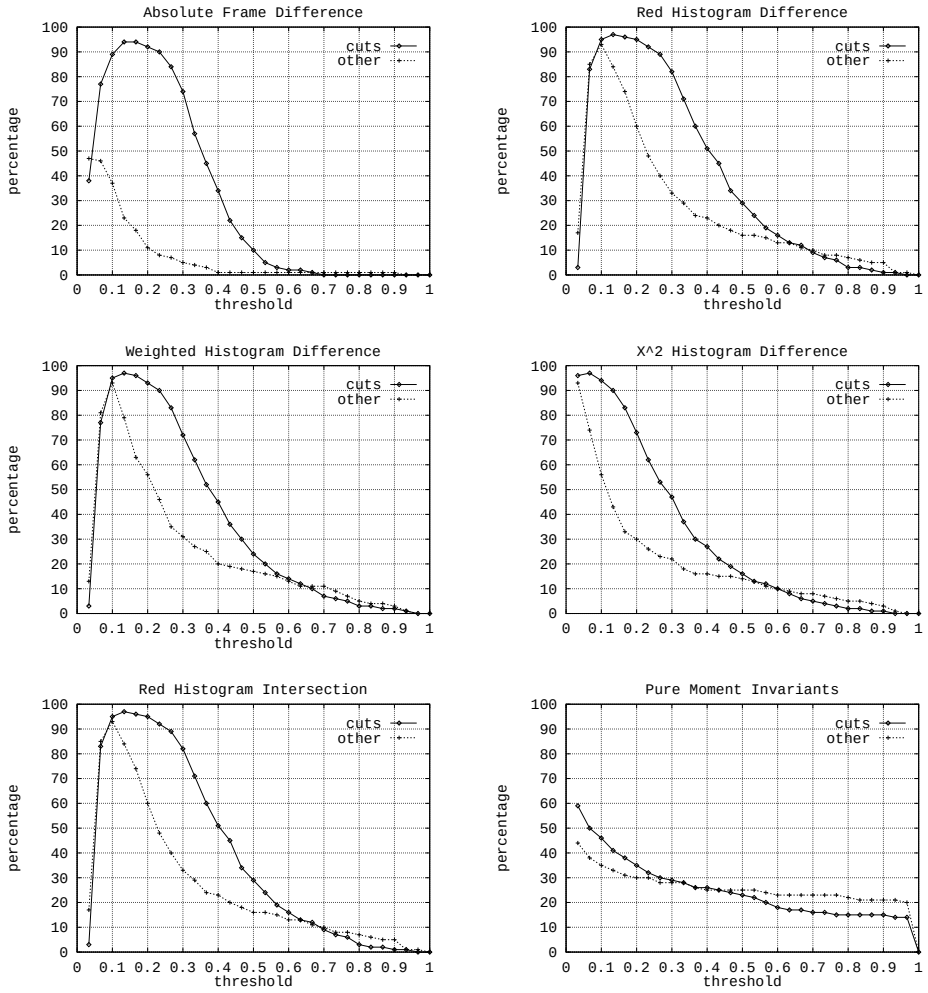
[8] K. Otsuji and Y. Tonomura, "Projection detecting filter for video and cut detection," *Proceedings of ACM Multimedia* (Aug. 1993) Anaheim, pp. 251-257.

[9] S. Smoliar and H. Zhang, "Content-based video indexing and retrieval," *IEEE Multimedia*, Vol. 2, pp. 62-72, 1995.

[10] M. Swain and D. Ballard, "Color indexing," *International Journal of Computer Vision*, Vol. 7, pp. 11-32, 1991.

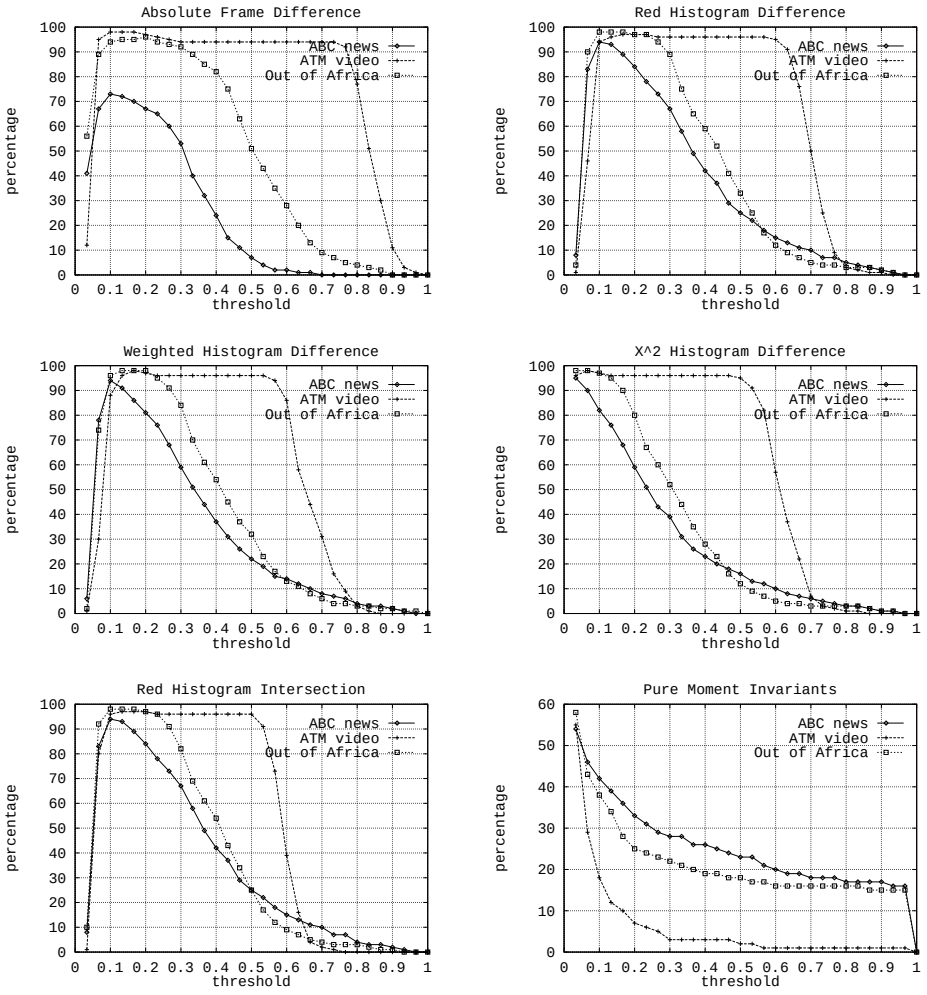
[11] M. Yeung, B-L Yeo, W. Wolf, and B. Liu, "Video browsing using clustering and scene transitions on compressed sequences," *IS&T/SPIE Multimedia Computing and Networking*, 1995.

[12] H. Zhang, S. Y. Tan, S. Smoliar, and G. Yihong, "Automatic parsing and indexing of news video," *Multimedia Systems*, 1995, Vol. 2, pp. 256-266.



Method	% Correct identifications of cuts	% Correct identifications of other effects
Range of pixel-value changes	97	91
Edge detection	92	91

Figure 3: Correctly identified cuts and other shot-transition effects for ABC news video.



Method	ABC News	ATM training	*Out of Africa*
Range of pixel-value changes	95	96	95
Edge detection	92	96	96

Figure 4: Correct shot-transition identifications across videos.