Computer Vision and Image Understanding **79,** 224–235 (2000)
doi:10.1006/cviu.2000.0858, available online at http://www.idealibrary.com on IDEAL®

Scene-Based Shot Change Detection and Comparative Evaluation

Loong-Fah Cheong

Department of Electrical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260
E-mail: eleclf@nus.edu.sg

Received October 4, 1999; accepted April 24, 2000

A key step for managing a large video database is to partition the video sequences into shots. Past approaches to this problem tend to confuse gradual shot changes with changes caused by smooth camera motions. This is in part due to the fact that camera motion has not been dealt with in a more fundamental way. We propose an approach that is based on a physical constraint used in optical flow analysis, namely, the total brightness of a scene point across two frames should remain constant if the change across two frames is a result of smooth camera motion. Since the brightness constraint would be violated across a shot change, the detection can be based on detecting the violation of this constraint. It is robust because it uses only the qualitative aspect of the brightness constraint—detecting a scene change rather than estimating the scene itself. Moreover, by tapping on the significant know-how in using this constraint, the algorithm's robustness is further enhanced. Experimental results are presented to demonstrate the performance of various algorithms. It is shown that our algorithm is less likely to interpret gradual camera motions as shot changes, resulting in a better precision performance than most other algorithms. However, its performance deteriorates under large camera or object motions. A twin-threshold scheme is proposed to improve its robustness.

1. INTRODUCTION

A key step for managing a large video database is to partition the video sequences into shots. Video partitioning makes the video data more manageable by imposing on it a hierarchy. It also forms the first step to understanding video content by dividing it into shots on which content analysis (e.g., scene, camera motion) can be performed.

1077-3142/00 $35.00

The conventional approach toward video partitioning treats the task as a 2-D image processing problem. Often, measures [15, 16] are proposed to capture the intuitive idea that the 2-D pixel intensity should undergo an abrupt change when there is a shot change. These measures include comparison of pixels, blocks, histograms, or DCT coefficients. It was found that all these approaches experience difficulties when the intensity changes are more gradual due to special editing effects, such as dissolve, fade-in, and fade-out. Such changes are often confused with those caused by smooth camera motions. A better approach is one where image intensity change caused by camera movements is taken into account. For instance, in [21], the optical flow field computed must exhibit certain patterns similar to that arising from a smooth camera motion. However, the criteria is based on simple patterns that are often violated in scenes with moving objects or in complex scenes where image flow exhibits patterns much more complicated than the templates.

The contribution of our method lies in using a rigorous physical constraint that is associated with the 3-D scene and motion, namely, the total brightness of a scene point across two frames should remain constant if the change across two frames is a result of smooth camera motion. Accordingly, since the 2-D image is formed from the projection of the 3-D scene onto the image plane, the pixel intensity of these 2-D images cannot change arbitrarily. The kind of intensity change expected from smooth camera motion is captured by the brightness constraint equation [8] and has been widely utilized in works dealing with optical flow. Thus, the problem of shot change detection becomes that of detecting when this scene-related constraint is violated.

Capturing the problem in this mathematical form has a twofold advantage. First, rather than dealing directly with the changes in image intensity across time, essential attributes are abstracted so that the total change in brightness across space and time are considered together. That is, pixel movements caused by camera motions are already taken into account. What this means is that images with textured surface and sharp intensity discontinuities undergoing motion will be better dealt with and not result in false alarms. This is in contrast to those simple methods where direct comparison of either pixels, regions, or histograms often reveals large changes.

Second, the assumptions under which the brightness constraint holds are clearly stated [8, 19]. From a computational standpoint, stating the assumptions has the advantage of allowing us to make judicious choice of the types of pixels or regions for analysis. For instance, from [19] it was known that in regions where the image intensity gradients are weak, the brightness constraint equation is most subject to the perturbations of other effects such as noise and occlusion. Avoiding analysis based on such regions would help reduce erroneous decisions and prevent confusion between shot changes and smooth camera transition. Importantly, we do not need the brightness constraint to hold absolutely at every point; such a requirement would be unrealistic in the face of noises in real images. We merely argue that the amount of violation caused by shot change and that caused by any inherent noise should be sufficiently different for discrimination.

This paper also presents a comparison of our algorithm with several other shot change detection methods, with the view to understand the limits faced by different classes of algorithms. The rest of the paper is organized as follows. In Section 2, we discuss related works in the literature. Section 3 presents our algorithm with technical details. Section 4 describes the algorithms selected for comparison and the implementation details. Experimental results are presented in Section 5 with discussions. Finally, the conclusions are given in Section 6.

2. RELATED WORKS

Numerous scene change detection algorithms have been proposed by several researchers and reviewed by [1, 9]. We characterize related works according to whether they explicitly take into account the image changes resulting from smooth camera operations.

Algorithms that are computing some kind of difference metric, whether it is based on pixels [16], block statistics [15], histograms [14], or DCT coefficients [20], all fall into the same category. They do not explicitly model the image difference caused by camera movements and are thus, strictly speaking, incapable of differentiating smooth camera operations from gradual scene transitions. While the use of more complex features such as image edges or histograms improves the situation, it will only relieve, but not remove, the problem.

Algorithms that are based on optical flow [2, 21] (computed from a pair of raw images) or on motion vectors [13] or macroblock data rates [12] (computed in the compressed domain) explicitly incorporate or compensate the image difference caused by smooth camera movements. The fact that the total change of image intensity, both in space and in time, is modeled should result in enhanced performance of these algorithms. However, most of these algorithms [12, 13, 21] did not exploit the characteristic of this change in a deeper manner. For instance, in the twin comparison approach [21], optical flow field computed was checked for its similarity with certain simple patterns expected to arise from typical smooth camera motions. Liu and Zick [12] used the ratio of the number of forward, backward, and bidirectional motion prediction vectors for B frames to detect cut. The results of such simple treatments are that these algorithms still face difficulties when the scene transitions are very gradual. On the other hand, [2] argued that smooth camera motions result in a flow that can be characterized by an affine model. This imposes a much stronger requirement on the properties of the scene change. The drawbacks are that it requires the scene in view to be locally planar and that it needs expensive computation to carry out the requisite segmentation. Compared to [2], our method exploits the fundamental property of the scene without imposing any domain specific assumption; it requires fewer assumptions and is therefore potentially more robust. However, optical flow techniques do rely on the assumption that the interframe displacements are small; we investigate the implication of this dependence in our experiments.

Recent literature has seen a number of model-based approaches [7, 13, 20]. The efficacy of these algorithms depends on the choice of certain parameters or thresholds which must be fine tuned. Furthermore, the assumption that the transition during special-effect edits is linear may not always be correct.

A number of researchers presented their evaluation of different approaches [3, 6, 10]. The recall and the precision ratio were often computed, the former evaluating the number of missed detections and the latter the number of false alarms:

$$Recall = \frac{detects}{detects + missed\ detects}$$

$$Precision = \frac{detects}{detects + false\ alarms}.$$

Kobla *et al*. [10] did a comparatively exhaustive comparison on special effect detection between their algorithm VideoTrails and four other algorithms. These four algorithms

were: (1) the plateau detection algorithm of Yeo and Liu [20]; (2) the variance curve approach of Meng *et al.* [13]; (3) the twin comparison approach of Zhang *et al.* [21]; and (4) the chromatic edit model approach of Song *et al.* [18]. The variance curve approach and the chromatic edit model approach were reported with poor recall performance, whereas the twin comparison approach and the plateau detection approach scored high recall but low precision.

3. PROPOSED METHOD

The shot detection method that we propose solves the problem of video segmentation based on the observation that there is a violation of the basic brightness constraint during shot changes. This observation is formulated more formally in the following.

3.1. Brightness Constraint Equation

Let $E(x, y, t)$ be the brightness of a point at image coordinates (x, y) and time t; E_x, E_y, and E_t denote the partial derivatives of image brightness with respect to x, y, and t, respectively. Then the aforementioned constraint can be expressed as

$$E_x u + E_y v + E_t = 0, \tag{1}$$

where $u = \frac{\partial x}{\partial t}$ and $v = \frac{\partial y}{\partial t}$ are the horizontal and vertical components of the optical flow, respectively. Optical flow is an approximation of the actual motion field. For instance, when the scene experiences a change in the level of illumination, the brightness constraint is violated. A nonzero optical flow field would be induced, even if there is no relative motion between the camera and the scene. In the current context, there will be a change in the scene illumination during cut, fade-in, fade-out, or dissolve. This violation of the brightness constraint would result in a "spurious" optical flow that has a haphazard appearance.

3.2. Optical Flow Under Shot Change

Equation (1) yields only the normal component of the optical flow (u, v), that is, the optical flow projected onto the normal direction (E_x, E_y). To obtain the full optical flow, we use the smoothness constraint proposed in [8].

The smoothness constraint assumes that neighboring points on objects have similar velocities and the optical flow field varies smoothly over the entire image. The problem of solving the optical flow then becomes that of simultaneously satisfying the brightness constraint and the smoothness constraint. This can be formulated as minimizing the error in the equation for the rate of change of image brightness,

$$E_b = u E_x + v E_y + E_t,$$

and the measure of the departure from smoothness in the optical flow,

$$E_c^2 = \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2. \tag{2}$$

The solution for the optical flow field is obtained by minimizing a weighted sum of E_b

and E_c

$$E^2 = \int\int \left(\alpha^2 E_c^2 + E_b^2\right) dx\, dy,$$

where α is the weighting factor determining the relative importance of the two terms E_b and E_c.

Since the brightness constraint is violated under shot changes, the two errors E_b and E_c may not be satisfied simultaneously. Even under smooth camera motion, the motion discontinuities that exist across object boundaries may give rise to conflicts between E_b and E_c. The result is that there will be error residues in both these situations. However, in the former, this phenomenon occurs globally over the whole image, whereas in the latter, it only occurs locally at object boundaries. Thus, we can use the sum of the error residue over the entire image as an indicator of shot changes. E, E_b, or E_c could have been used in the measure, depending on the value of α used. For the conventional value of $\alpha = 0.5$, we found that E_c is the most effective.

4. IMPLEMENTATION

This section provides a brief description of the five algorithms that are included in this paper for comparison. The reader is referred to the cited papers for more details.

Our algorithm. We have implemented our algorithm for shot detection as defined in the previous section. Our method first preprocessed the images with Gaussian smoothing, using a 5×5 spatial filter and a temporal filter of 5 time units. Optical flow analysis was then carried out with the value of α set to 0.5. A threshold T_i on the intensity gradient was set to ensure that only sufficiently textured areas are considered. T_c was another threshold beyond which the error residue in E_c was considered significant. The values used for these thresholds were rather conventional and were not tuned in this experiment. The following was then defined,

$$n_1(x, y) = \begin{cases} 1 & \text{if } |E_c| > T_c \text{ and } \left|E_x^2 + E_y^2\right| > T_i \\ 0 & \text{otherwise} \end{cases}$$

$$n_2(x, y) = \begin{cases} 1 & \text{if } \left|E_x^2 + E_y^2\right| > T_i \\ 0 & \text{otherwise} \end{cases}$$

$$D = \frac{\sum_y \sum_x n_1(x, y)}{\sum_y \sum_x n_2(x, y)} \times 100\%,$$

where D measured the departure of the optical flow from smoothness, and the partial derivatives in E_c were approximated from the discrete set of optical flow measurements in a 5×5 spatial window. We declared a shot change if D is greater than $T_1 = 12$. Furthermore, since fast movements often generate a series of closely spaced false alarms, we devised a rule to reduce such false alarms. If a series of detects are close (less than 20 frames apart), retain them only if D is greater than a second threshold $T_2 = 2 \times T_1$. This amounts to a thresholding rule that is context dependent and is similar in idea to the twin comparison method.

Plateau detection [20]. In the plateau detection algorithm, abrupt changes and gradual transitions are detected separately. An abrupt scene change is declared if there is a sharp

peak in the D_i plot, where D_i is simply the DC difference between frame i and frame $i+1$. To ensure sharpness of the peak, the peak value must be n times the second largest maximum in a symmetric sliding window. After our tuning experiment, n was chosen to be 1.5, and the size of the window was 20. Gradual transitions are then detected by looking for plateaus on the D_i^k plot. The DC difference D_i^k is computed by comparing every frame to the following kth frame. In our experiment, k was chosen to be 20. Two criteria are then used by the authors Yeo and Liu to detect the plateau on the difference plot. The first criterion checks that there is little variation on the plateau. In our implementation, we allowed for a 20% variation within a symmetric window of size 5. The second criterion checks that the plateau stands out by the following: $D_i^k \geq l \times D_{i-k/2-1}^k$ or $D_i^k \geq l \times D_{i+k/2+1}^k$. We used the value of $l = 2.8$; we also compared D_i^k with D_{i-k-1}^k and D_{i+k+1}^k instead of with $D_{i-k/2-1}^k$ and $D_{i+k/2+1}^k$, respectively. This modification was also suggested by [10].

Variance curve [13]. In the variance curve approach, an abrupt change is declared if there is large change in the variance plot (in our implementation, a 35% change in the variance σ^2). The variance plot is then examined for downward parabolic curves to detect dissolves. The two peaks that bound a valley and the valley are first located. To qualify as a candidate for gradual transition, the distances between the peaks and the valley must each be at least four frames. If this criterion is satisfied, and if either of the drops in variance from the peaks to the valley is more than 25%, a dissolve is declared. In addition, as suggested by [10], if the average variance difference $|\triangle\sigma^2|$ between the two peaks is greater than that on either end of the peaks, a dissolve is also declared. We used a local window of 10 frames before the left peak and 10 frames after the right peak for the $|\triangle\sigma^2|$ computation, and we required that the difference be at least 2.5 times.

Twin comparison [21]. The implementation for the twin comparison approach is quite straightforward. The twin thresholds T_b (used for break detection) and T_s (used for special effect detection) were respectively 25,000 and 15,000.

Chromatic edit model [18]. Finally, in the chromatic edit model approach, the first partial derivative of the intensity with respect to time is computed. If there is a large change (500%) in this first derivative, a cut is declared. The second partial derivative of the intensity with respect to time is then computed. If it is small compared with the first partial derivative, a dissolve is declared. As found out by [10], a large value is required for the test to work. We used 0.9 as the fraction value for the threshold.

Error metric and others. We chose recall and precision as the evaluation criteria. In particular, a gradual transition was considered correctly detected if any of the frame of the transition was marked as a shot boundary. Conversely a missed dissolve was counted as one miss, irrespective of the duration of the effect. Often, a fast action followed immediately after a scene cut, resulting in the reported shot boundary being displaced from the true cut location, in which case we counted as a missed cut and a false alarm. On the other hand, if a long dissolve was detected as multiple transitions, we counted one of them as a correct detect, with the rest treated as false alarms. Furthermore, since various algorithms (e.g., in the plateau detection algorithm) ruled out the possibilities of two scene changes within a short duration, we applied a similar rule to all other algorithms where this step was not explicitly taken. In particular, if there is a series of transitions $t_1, t_2, \ldots, t_n$, each separated by less than 10 frames, retain only t_1 and t_n. We found that this rule was especially important in helping the variance curve approach and the twin comparison approach to reduce the number of false alarms arising from fast motions.

5. EXPERIMENTS AND RESULTS

This section describes the two sets of experiments conducted, the corresponding test sets, and the results obtained.

5.1. Experiment One

The purpose of this set of experiments is twofold. First, we want to illustrate the performance of these algorithms under various challenging scenarios, such as long dissolves and fast motions. Different algorithms are likely to work well in different scenarios and it is important to understand in which domain a given algorithm does work well. Such performance characteristics are not always obvious from the recall and precision figures reported from typical test sets. The second objective of this set of experiment is to fine tune the various parameters involved. The thresholds have been chosen such that high recall ratio is given more priority, without penalizing precision unduly if possible.

A small number of real-world video clips (about 4100 frames) extracted from VHS tapes were used. They were comprised of clips from a nature documentary showing the landscape and the fauna of the Okavango Delta, a documentary commemorating the life of Princess Diana, and a recorded clip depicting soccer action. These videos were quite noisy and contained a mix of fast image motions and long dissolves (especially in the Diana sequence).

Some examples from the Diana clip and the soccer clip are illustrated in the top row of Fig. 1. The Diana clip typifies documentaries of this genre: there are many slow transitions lasting more than two seconds. Other likely problems of the clip are that part of it depicts fast moving objects and that the first 500 frames of this sequence are very noisy. The soccer clip contains many fast camera and object motions; some of the close-ups are particularly problematic. This clip also contains some sport-style edits such as wipes.

The second and third rows of Fig. 1 present the performances of the various algorithms for the Diana clip and the soccer clip respectively. The vertical axes of the plots correspond to the metrics used in the respective methods, although *SD*" of the twin comparison plot has been somewhat modified from the original metric to facilitate plotting. It is defined as the accumulated difference between the current frame and the potential starting frame of a transition once such a potential starting frame has been identified and has not been dropped; otherwise it is just the difference between consecutive frames defined by the difference metric. Dotted lines in the figures correspond to the various thresholds used in these algorithms. The thresholds shown in the plots for our algorithm correspond to T_1 and T_2 mentioned above, whereas those shown in the twin comparison plots correspond to the twin thresholds T_b and T_s. In other cases, the final decision for detection is not obvious from the plots. Thus, all graphs were annotated with the following letters: we use C, S, F, and M to denote cut detect, special-effect detect, false alarm, and miss, respectively.

The results of the Diana sequence showed that the model-based methods, including the plateau detection, the variance curve, and the chromatic edit approach, performed better than the rest in picking up dissolves longer than two seconds. On the other hand, as evidenced by the plots for the soccer sequence, by adopting appropriate models, the plateau detection algorithm and the chromatic edit algorithm achieved better immunity against false alarms arising from fast motions.

The twin comparison and our approach adopt a twin-threshold scheme to relieve the trade-off problem involved in using a single threshold. While such a scheme was effective

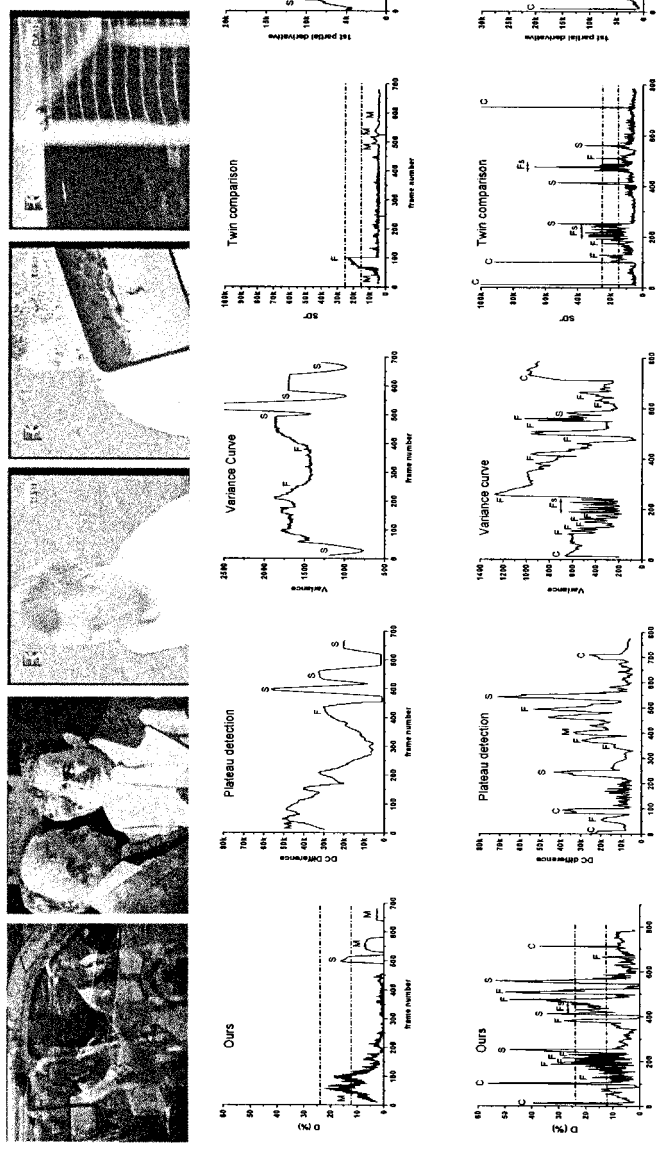

FIG. 1. First row from left to right: Diana sequence: frames 30 (dissolve coupled with fast motion) and 550 (dissolves); soccer sequence: frames 196 (close up), 245 (fold-up), and 508 (shoot). Second and third rows illustrate the performance of the various algorithms for the Diana clip and the soccer clip respectively. The dotted lines shown in the plots for our algorithm and the twin comparison algorithm correspond to the twin thresholds in both algorithms. The letters *C*, *S*, *F*, and *M* are used to denote cut detect, special-effect detect, false alarm, and miss, respectively (F's refers to multiple false alarms). For the twin comparison plot, refer to the text for the definition of its vertical axis, *SD*".

to filter away some of the false alarms arising from fast motions (see the results of the soccer sequence), the problem was not totally removed. Without the aid of a model, these methods were still fundamentally limited in detecting dissolves of longer duration, as can be seen by the results of the Diana sequence. The virtue of these two methods lies in their simplicity, which means that they are less likely to fall prey to invalid assumptions such as linearity. In the Diana sequence significant, noise in the same sequence also resulted in a comparatively higher number of false alarms for most linear-model-based approaches. These effects are in general more difficult to control or to compensate for.

5.2. Experiment Two

In the second set of experiments, the tuned algorithms were applied to a test set comprising of a large number of video clips (about 30,000 frames with 233 edits). They were drawn from wide-ranging categories, including documentaries, news, commercials, music videos, sports, and action movies. Table 1 summarizes the characteristics of these different clips, with a brief description of the troublesome elements in these clips. For instance, special effect edits such as wipes, additive dissolves, and fold-ups are often used in sports videos. Commercials often have edits closely spaced together, and the duration of the edits are often very short. On the other hand, parts of the documentaries have dissolves as long as 180 frames (6 s). There are also many video clips where the assumptions underlying the various algorithms are violated. For instance, action movies and sport clips invariably contain large interframe camera or object movements exceeding the range required of optical flow computation. Music videos and commercials have scenes that violate the assumption

TABLE 1
Different Classes of Test Video

Class	Description	Types of gradual transitions	Fast movements	Other problems
Sports	4 clips (2 cricket and 2 soccer)	Dissolves, wipes, and fold-ups	Tracking activities; close-up on players;	Flashes
Commercial	4 clips (Discovery channel, Pampers, Lifestyle, cough drops)	Large number of short dissolves, graphics overlay	Moderate	Rapid rhythm of edits (15 frames apart), burning flames
Music video	2 clips (1 on-stage and 1 outdoor)	Dissolves, fade-ins and fade-outs	Singers' movements	Rhythm of edits mixed; floodlight
News	2 clips (BBC news on Chechnya war)	Short dissolves		Footage on war is noisy; cloud shadows resulting in intensity changes
Documentary	3 clips (Diana, Okavango Delta, and Serengeti Plain)	Many long dissolves (1–6 s), fade-in, fade-out.	Animals' movements	Footage on Diana is noisy
Action movie	2 clips (movie "Speed" and TV series "Martial Law")	None	Many fast actions and extreme close-ups	Fast actions resulting in displaced shot boundaries

TABLE 2
Performance of the Various Algorithms

	Ours			PL			VC			TC			CE		
	C	F	M	C	F	M	C	F	M	C	F	M	C	F	M
Sports	27	23	0	25	17	2	17	35	10	27	15	0	20	7	7
Commercial	54	5	8	54	5	8	42	9	20	53	16	9	56	14	6
Music video	57	2	6	58	5	5	38	8	25	60	6	3	59	4	4
News	23	6	0	21	14	2	12	12	11	22	13	1	20	16	3
Documentary	5	3	20	14	24	11	24	19	1	7	8	18	14	11	11
Action movie	29	36	4	30	6	3	23	82	10	30	26	3	26	24	7
Total	195	75	38	202	71	31	156	165	77	199	84	34	195	76	38
Recall %		83.7			86.7			67.0			85.4			83.7	
Precision %		72.2			74.0			48.6			70.3			72.0	

of constant illumination (flashes). Finally, there are shot changes where linear assumption fails, resulting in difficulties for algorithms assuming linear transition. In sum, the test set is constructed to be as comprehensive as possible.

The results of the second experiment are tabulated in Table 2 (PL, plateau detection; VC, variance curve; TC, twin comparison; CE, chromatic edit). It shows that our approach, the plateau detection approach, the twin comparison approach, and the chromatic edit approach gave satisfactory performance, whereas the variance curve approach did not yield good results.

Of the model-based approaches, the plateau detection method performed very well. It is capable of picking up long edits and yet its immunity to false alarms when there are fast movements is unsurpassed (see its performance under the sports and action movies genres). It is difficult to further improve on the choice of the plateau parameters k and l. For instance, it is hard to set an optimal value of k for all sequences. A large k tends to merge closely-spaced edits (such as those found in commercials) together. Conversely a small k tends to split edits of long duration into multiple transitions (such as those found in documentaries, thus its high false alarm rate).

The variance curve approach did not perform well. While it has the best recall rate in the documentary genre (with long dissolves), it tends to miss a lot of transitions especially if the scenes do not contain enough variance (e.g., dim scenes in music videos, scenes of a cricket field in sport). Furthermore, the peak and valley locations are very sensitive to perturbances such as fast actions. Peaks and valleys occur at different scales; some of these correspond to transition events and some do not. It was found difficult to have an optimal smoothing so that only the "desirable" events are picked up. Thus its performance (both recall and precision) declined drastically in the sport and action movies genres.

The chromatic edit approach has slightly inferior performance compared to the plateau detection approach. The poorer precision of the chromatic edit approach could be attributed to the proposed test for gradual transitions. A large fraction value needs to be set in order to pick up gradual transitions. The accompanying increase in false alarm rate is not unexpected. However, this increase in false alarms could not be strictly attributed to a sole factor such as fast motions. These false detects were not clustered together and were therefore more difficult to compensate for.

The results of the twin comparison approach and ours were quite comparable. They were also slightly inferior to that of the plateau detection approach. Both false alarm rates increased chiefly with fast actions. However, this is amenable to the treatment of the twin threshold concept (since they are clustered together), with the result that the deterioration is graceful. In particular, if the camera–object motions are not of the large foreground types, both algorithms did not yield significant false alarms. In fact, if we exclude the sports and action movie categories, our algorithm has the lowest false alarm rates among all the algorithms tested. Unfortunately, close-up shots of fast actions such as those found in action movies and sports seemed to pose severe challenge for our algorithm. Furthermore, both algorithms still faced problems in picking up dissolves longer than two seconds, as evidenced by the high number of misses in the documentary genre. There is a weak linear assumption implied in the twin comparison method. If an edit has long duration, the consecutive difference value may fall below the lower threshold T_s. It can result either in a miss or in multiple detects. This problem was discussed in Zhang's paper and the authors suggested setting a tolerance value that allows a number of consecutive frames with low difference value before rejecting the transition candidate.

Illumination changes arising from camera flashes in the sport videos did not cause false alarms as the effect was local (i.e., the flashes occurred in a long shot of the spectator scene). Illumination changes arising from natural causes such as passing clouds, while global in effect, are much more gradual. They are apt to be picked up as false alarms by the model-based approaches, which are more sensitive to such gradual changes lasting over a long duration. This accounts for the higher number of false alarms reported by the model-based approaches in the news clips where overcast weather produced such global changes.

For applications where real time processing is required, our method is at a disadvantage due to the significant amount of processing involved in optical flow computation. The time taken by our nonoptimized algorithm for a 352×288 image is about 30 s per frame. While various fast parallel methods of optical flow computation have been proposed [4, 11], they require varying amounts of dedicated hardware. For instance, the algorithm in [4] requires parallel hardware computation of the Laplacian of Gaussian of the images, with the rest of the computation performed with common desktop hardware.

From the results of these experiments, we can make the following observations. As far as our algorithm was concerned, it seemed that the degradation in performance caused by image noise and moderate motions was largely well controlled. The well-known flow inaccuracies caused by the Horn and Schunck's method did not affect the validity of our results, thus corroborating our claim that only a qualitative aspect of the optical flow is used. However, if the scenes contained large interframe displacements (typically greater than 10 pixels in sports and action movies), the assumption underlying optical flow computation was severely violated, resulting in a rapid increase in the false alarm rate. However these false alarms were mostly closely clustered together, and thus to a large extent can be controlled by a context-dependent twin-thresholding rule.

6. CONCLUSION

We have proposed an algorithm that utilizes a basic constraint associated with 3-D scene points—the brightness constraint. As it only needs partial information about the scene—detecting a scene change rather than estimating the scene itself—it does not need accurate optical flow information and is thus robust. By using this basic constraint, our algorithm

is capable of handling a diverse range of situations such as textured images undergoing motion and nonlinear shot transitions. However, when dealing with extreme situations, such as very large motions or transitions lasting more than two seconds, our algorithm faces limitations. Among the algorithms compared, the plateau detection approach, the twin comparison approach, the chromatic edit approach, and our approach have quite comparable performances, as far as this particular test set is concerned. In general, the good model-based approaches are more successful in picking up transitions lasting more than two seconds and are less susceptible to false alarms arising from fast motions. For the non-model-based approaches, a twin-thresholding scheme seems to be required in order to overcome the trade-off involved in using a single threshold.

REFERENCES

1. G. Ahanger and T. D. C. Little, A survey of technologies for parsing and indexing digital video, *J. Visual Comm. Image Representation*, special issue on Digital Libraries **7**, 1996, 28–43.
2. P. Bouthemy and F. Ganasia, Video partitioning and camera motion characterization for content-based video indexing, in *Proc. ICIP* **1**, 1996, 905–908.
3. J. S. Boreczky and L. A. Rowe, Comparison of video shot boundary detection techniques, *SPIE* **2670**, 1996, 170–179.
4. A. K. Chhabra, Real-time computation of optical flow along contours of significant intensity change, *Real-time Imaging* **3**, 1997, 87–99.
5. R. M. Ford, Quantitative comparison of shot boundary detection metrics, *SPIE* **3656**, 1999, 666–676.
6. U. Gargi, R. Kasturi, and S. Antani, Performance characterization and comparison of video indexing algorithms, in *IEEE Conference on Computer Vision & Pattern Recognition, 1998*, pp. 559–565.
7. A. Hampapur, R. Jain, and T. Weymouth, Production model based digital video segmentation, *Multimedia Tools Appl.* **1**, 1995, 9–46.
8. B. K. P. Horn and B. G. Schunck, Determining optical flow, *Artificial Intelligence* **17**, 1981, 185–203.
9. F. Idris and S. Panchanathan, Review of image and video indexing techniques, *J. Visual Comm. Image Representation* **8**, 1997, 146–166.
10. V. Kobla, D. DeMenthon, and D. Doermann, Special effect edit detection using VideoTrails: A comparison with existing techniques, *SPIE* **3656**, 1999, 302–313.
11. J. J. Little, H. H. Bülthoff, and T. Poggio, Parallel optical flow computation, in *Proceedings Image Understanding Workshop, Los Angeles, CA, February 1987*, Morgan Kaufmann, San Mateo, CA, pp. 915–920.
12. H. C. Liu and G. L. Zick, Scene decomposition of MPEG compressed video, *SPIE* **2419**, 1995, 26–37.
13. J. Meng, Y. Juan, and S. F. Chang, Scene change detection in a MPEG compressed video sequence, *SPIE* **2419**, 1995, 14–25.
14. N. V. Patel and I. K. Sethi, Video shot detection and characterization for video databases, *Pattern Recognition* **30**, 1997, 583–592.
15. I. K. Sethi and N. Patel, A statistical approach to scene change detection, *SPIE* **2420**, 1995, 26–37.
16. A. Seyler, Probability distributions of television frame differences, *Proc. IEEE* **53**, 1965, 355–366.
17. S. Shahraray, Scene change detection and content-based sampling of video sequences, *SPIE* **2419**, 1995, 2–13.
18. S. M.-H. Song, T.-H. Kwon, and W. M. Kim, On detection of gradual scene changes for parsing of video data, *SPIE* **3312**, 1997, 404–413.
19. A. Verri and T. Poggio, Motion field and optical flow: Qualitative properties, *IEEE Trans. Pattern Anal. Mach. Intelligence* **11**(5), 1989, 490–498.
20. B. L. Yeo and B. Liu, Rapid scene analysis on compressed video, *IEEE Trans. Circuits Systems Video Technol.* **5**, 1995, 533–544.
21. H. Zhang, C. Y. Low, and S. W. Smoliar, Video parsing and browsing using compressed data, *Multimedia Tools Appl.* **1**, 1995, 89–111.