SHOT BOUNDARY DETECTION FROM VIDEOS USING ENTROPY AND LOCAL DESCRIPTOR

Junaid Baber, Nitin Afzulpurkar, Matthew N. Dailey, and Maheen Bakhtyar

School of Engineering and Technology
Asian Institute of Technology
Pathumthani, Thailand.
Email: {junaid.j.baber, nitin, mdailey, maheen.bakhtyar}@ait.ac.th

ABSTRACT

Video shot segmentation is an important step in key frame selection, video copy detection, video summarization, and video indexing for retrieval. Although some types of video data, e.g., live sports coverage, have abrupt shot boundaries that are easy to identify using simple heuristics, it is much more difficult to identify shot boundaries in other types such as cinematic movies. We propose an algorithm for shot boundary detection able to accurately identify not only abrupt shot boundaries, but also the fade-in and fade-out boundaries typical of cinematic movies. The algorithm is based on analysis of changes in the entropy of the gray scale intensity over consecutive frames and analysis of correspondences between SURF features over consecutive frames. In an experimental evaluation on the TRECVID-2007 shot boundary test set, the algorithm achieves substantial improvements over state of the art methods, with a precision of 97.8% and a recall of 99.3%.

***Index Terms*—** Shot boundary detection, SURF, Run length encoding

1. INTRODUCTION

Videos are sources of education, entertainment and information, and video databases are rapidly increasing in number and size. As the volume of publicly available video data increases, it is becoming more difficult to index and retrieve them from their databases. Therefore, there is a need for robust, efficient, and accurate video indexing algorithms. One of the most common approaches to video indexing involves breaking the video into *shots* and storing a representative frame for each shot. A video shot is an uninterrupted sequence of frames depicting a single scene or event, and segmenting a video into shots means identifying the boundaries between consecutive shots.

There are two types of shot boundaries, abrupt and gradual. These types of boundaries are also known as hard cuts and soft cuts, respectively. Abrupt shot boundaries occur when the scene changes immediately between two frames, e.g., when the camera focus changes from one person to another during a conversation. Gradual shot boundaries, on the other hand, involve gradual scene changes over several frames. Gradual shot boundaries often occur at the beginning or end of television shows, advertisements, and movies; the effects include fade in, fade out, and dissolve.

Abrupt boundary detection is relatively easy compared to gradual boundary detection because the difference between two consecutive frames at an abrupt shot boundary can be easily detected. Let $\{f_i\}_{i\in 1,\ldots,n}$ be the frames of a video. f_i can be considered an abrupt shot boundary if the similarity of f_i and f_{i+1} is low. Gradual shot boundary detection, on the other hand, requires comparison of frame f_i to frame f_{i+k} for some unknown $k > 0$.

In the literature, many algorithms for shot boundary detection have been proposed. Some of the algorithms are extremely effective for particular types of data including news, sports, dramas, or advertisements. Zhang et al. [1] use a very simple frame-to-frame pixel intensity difference measure

$$D(f_i, f_{i+1}) = \frac{\sum_{x=1}^{W}\sum_{y=1}^{H}|f_i(x,y) - f_{i+1}(x,y)|}{W \times H},$$

where $W \times H$ is the number of pixels in the image and $f_i(x,y)$ is the gray level intensity of the pixel at coordinates (x,y) in frame i. The method signals an abrupt shot boundary at f_i when $D(f_i, f_{i+1})$ is above some threshold T. The main limitation of this approach is that it is sensitive to camera and object movement [2].

One solution to the problem of a moving camera or object is, rather than comparing individual pixel intensity differences, to use a difference measure comparing histograms derived from successive frames [3, 4, 1], in particular,

$$D(f_i, f_{i+1}) = \sum_{k=1}^{N} |\mathrm{Hist}_{f_i}(k) - \mathrm{Hist}_{f_{i+1}}(k)|,$$

where $\mathrm{Hist}_{f_i}(k)$ is the k-th bin of the gray level histogram for frame f_i and N is total number of bins. Some of the limitations of this approach are that two distinct images can be represented by same histogram and that it does not detect gradual shot boundaries.

978-1-4577-0274-7/11/$26.00 ©2011 IEEE

Cernekova et al. [5] propose a shot boundary detection method based on mutual information and joint entropy between successive frames for a sports dataset. At abrupt boundaries, the mutual information is low. Joint entropy is useful for detecting fades; the joint entropy will be high for an extended period of time during a fade-in, as visual intensity gradually increases, or during a fade-out, as visual intensity gradually decreases. Chavez et al. [6] propose supervised learning with support vector machines (SVMs) to separate abrupt boundaries from non-abrupt boundaries. To capture information about illumination changes and fast motion, they calculate a dissimilarity vector incorporating a rich set of features, including Fourier-Mellin moments, Zernike moments, and color histograms (RGB and HSV). After the dissimilarity vector is input to the SVM to detect abrupt boundaries, illumination variance and an average gradient over the image are used to detect gradual boundaries. Ling et al. [7] propose another learning algorithm based on SVMs comprised of three steps. In the first step, frames unlikely to be shot boundaries, due to smooth changes, are removed. In the second step, features including intensity differences, differences in vertical and horizontal edge histograms, and differences between HSV color histograms are extracted then classified by a SVM to detect abrupt boundaries. In the third step, gradual boundaries are detected using temporal multi-resolution analysis.

We propose a method that is not only able to detect abrupt boundaries accurately but is also able to detect fade boundaries with high accuracy. Fade boundaries are first detected indiscriminately based on temporal changes in the entropy of the pixel intensity across each image, then analysis of SURF feature correspondences across candidate boundaries is used to reject likely false positive boundaries. The method achieves high precision and recall on the TRECVID 2007 test data set.

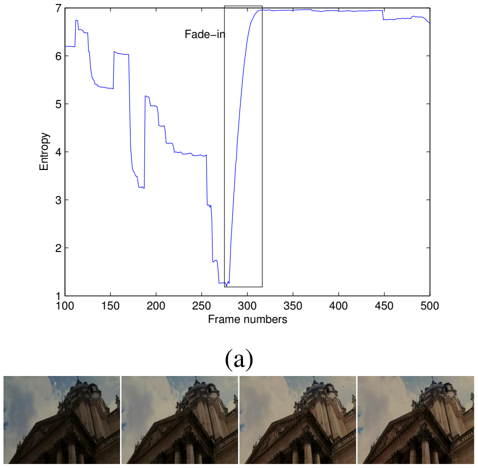
2. BACKGROUND AND DEFINITIONS

2.1. Entropy

Entropy measures the uncertainty of a random variable. Let X be a discrete random variable over events $A_X = \{y_1, \ldots, y_N\}$ with associated probabilities $P_X = \{P_1, \ldots, P_N\}$. The entropy of X is

$$E_X = -\sum_{y \in A_X} P_y \log P_y$$

We use the entropy of an image's gray scale intensities to track image changes over time. We assume a discrete random variable I with values from 0 to 255 and treat an image's pixel intensities as i.i.d. samples from an unknown discrete distribution P_I over I. To estimate P_I, given an input image, we first calculate a histogram H_I of the pixel intensities then normalize.



(a)

(b)

Fig. 1. Gradual shot boundary detection by entropy. (a) Gradual increase in entropy at a fade-in. (b) A sub-sequence of the frames in which the fade-in boundary is detected. Frames are taken from the *The Pink Panther* (2006).

2.2. Local keypoint descriptors

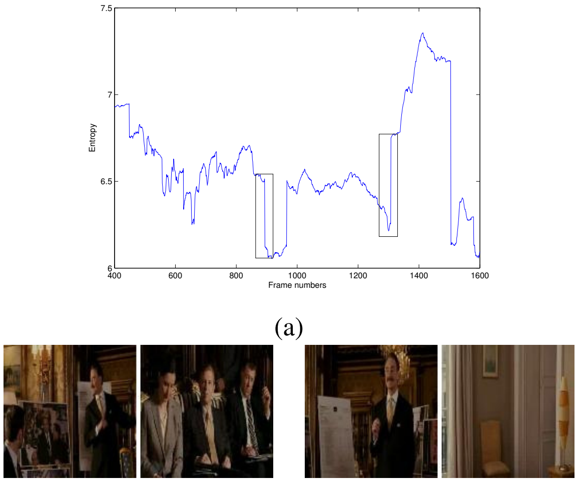
As previously mentioned, we detect candidate shot boundaries fairly indiscriminately. We then use the Speeded Up Robust Features (SURF) [8] algorithm to find point correspondences across a candidate shot boundary, treating as a false positive any candidate boundary across which the point correspondences are consistent. The SURF algorithm identifies keypoints in an image then provides a 64-element vector describing the texture around each keypoint, normalized for rotation and scale. The descriptors can be used for fast and robust point matching between two images under scale, rotation, noise, illumination changes, and changes in a cluttered background. Interested readers are referred to the original work for details.

3. SHOT BOUNDARY DETECTION

In the proposed method, shot boundaries are extracted from videos using frame entropy and SURF descriptors. Fade boundaries are found by detecting changing patterns of entropy during fade effects. Abrupt boundaries are detected by difference of entropy in adjacent frames, and SURF is also used in abrupt boundary detection. The frequency of occurrence of abrupt shot boundaries in videos is higher than the occurrence of gradual shot boundaries. Therefore, abrupt shot boundaries are the main focus of the paper.

3.1. Gradual shot boundaries

In a fade effect, there is change of illumination either from a dark image to a brighter image or a bright image to a darker



(a)

(b)

(c)

Fig. 2. Abrupt boundary detection using pixel intensity entropy changes. (a) Large changes in entropy. (b,c) Shot boundaries corresponding to changes shown in panel (a). Frames are from *The Pink Panther* (2006).

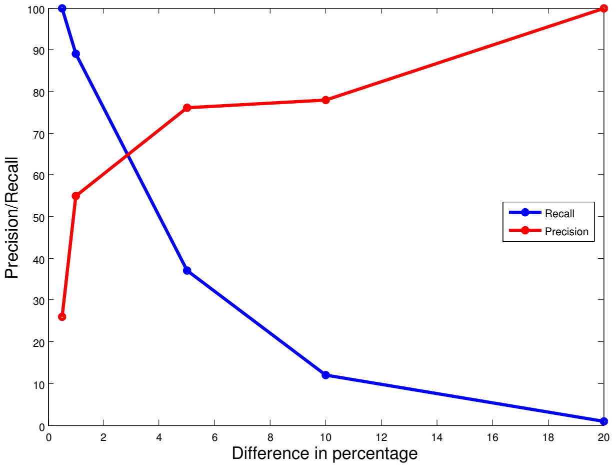


Fig. 3. Precision and recall at different pixel intensity entropy difference threshold values. The curves are for *The Pink Panther* (2006).

image. The entropy of a dark image is very low, close to zero. During fade-in and fade-out effects, entropy gradually increases or decreases. For example, in the case of a fade-in boundary, the pixel intensity entropy will gradually increase, as shown in Figure 1.

To find increases or decreases in the pixel intensity entropy, we use a *fade signature*. For a video V, represented by entropy measurements $V = \{f_{e_1}, f_{e_2}, \ldots, f_{e_n}\}$, where f_{e_i} is the entropy of the ith frame and n is total number of frames in video, the fade signature is $FS = \{u_1, u_2, \ldots, u_{n-1}\}$, where u_i is defined as

$$u_0 = 0$$
$$u_i = \begin{cases} 1 & f_{e_i} > f_{e_{i+1}} \\ 0 & f_{e_i} < f_{e_{i+1}} \\ u_{i-1} & f_{e_i} = f_{e_{i+1}} \end{cases}$$

FS is a binary sequence with 0 indicating that entropy is increasing and 1 indicating that entropy is decreasing. We perform run length encoding (RLE)[1] on the FS sequence to obtain a sequence of pairs in which the first element is a value in the FS sequence and the second element is the number of consecutive times the value appears in the sequence. For example, given the FS sequence $\{0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0\}$, the RLE is $\{(0, 2), (1, 3), (0, 1), (1, 1), (0, 5)\}$. We enrich the RLE with the minimum entropy of the frames in each run to obtain a sequence of triples $R = \{r_1, r_2, \ldots, r_m\}$, where $r_j = (a_j, b_j, c_j)$, with a_j either 0 or 1, b_j the length of the run of a_j, and c_j the minimum entropy of the b_j frames in the run. If the length $b_j > T_f$ and $0 < c_j \leq T_e$, we consider the run to be a fade effect with a_j indicating whether the run is a fade-in or a fade-out. The constraint on c_j ensures that the entropy has either decreased to or increased from a low entropy. Without this constraint, many false alarms are detected as dissolves, gradual appearances of objects, and so on. Once all fades are detected by RLE on FS, we can easily find the boundaries of fades. For the fade boundary[2], cumulative sum of all frequencies $\sum_{i=1}^{j} b_i$ is taken.

The value of T_f can be computed experimentally or by using some prior information and the number of frames per second. Since fade effects last for a few seconds, T_f can be set to the number of frames expected to occur during the minimum expected fade duration.

3.2. Abrupt shot boundaries

In an abrupt shot boundary, the camera perspective changes instantly from one frame to the next. This reliably causes a big difference in pixel intensity entropy as shown in Figure 2. Whenever the difference in entropy between two successive frames is above a threshold T, we declare a shot boundary.

The threshold T can be set experimentally to give high recall or high precision, as shown in Figure 3. However, while low values of T give high recall, they also lead to low precision due to many false alarms occuring with changes in a single shot. Examples are shown in Figure 4. Conversely, high values of T give high precision but low recall. Some examples of shot boundaries with relatively small entropy changes are shown in Figure 5.

To overcome the problems mentioned above, we use two steps, as shown in Figure 6. In the first step, shot boundaries are found by comparing the difference of entropy of two adjacent frames with threshold T_1. High values of T_1 are used

[1] http://mathworld.wolfram.com/Run-LengthEncoding.html

[2] Actual location of fade in video

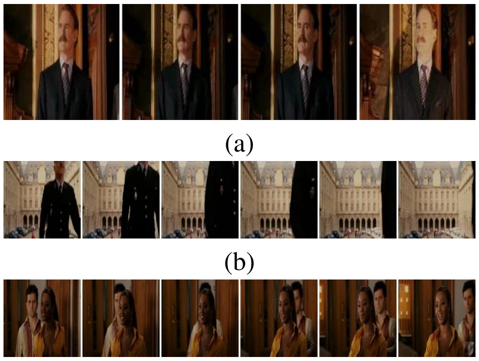
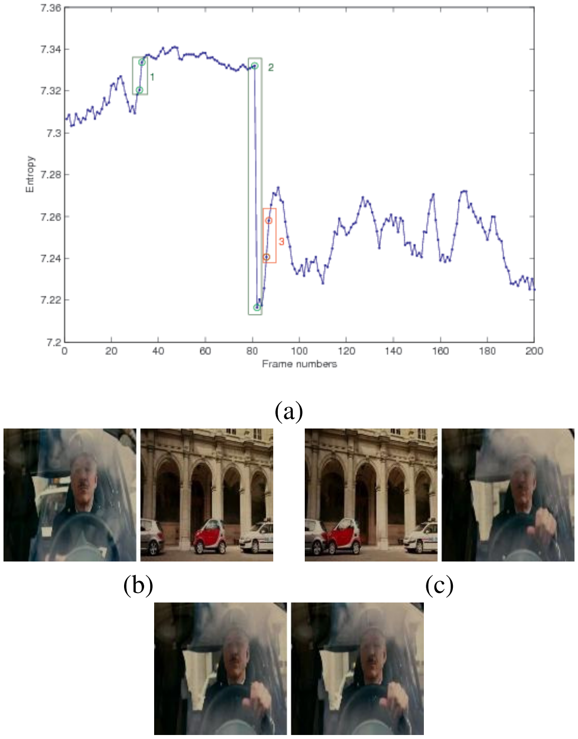

(a)
(b)
(c)

Fig. 4. False detection of shot boundary using entropy method. a) A change in light intensity (a flash). b) Gradual disappearance of an object from a scene. c) Significant motion in the scene.



(a)
(b)
(c)
(d)

Fig. 5. Examples of problems with entropy difference based shot boundary detection. a) Three possible abrupt shot boundaries, where (1) and (2) are true shot boundaries but (3) is not a shot boundary. The difference of entropy at (3) is greater than (2) but less than (1). (b) Adjacent frames for point (1). (c) Adjacent frames for point (2). (d) Adjacent frames for point (3).

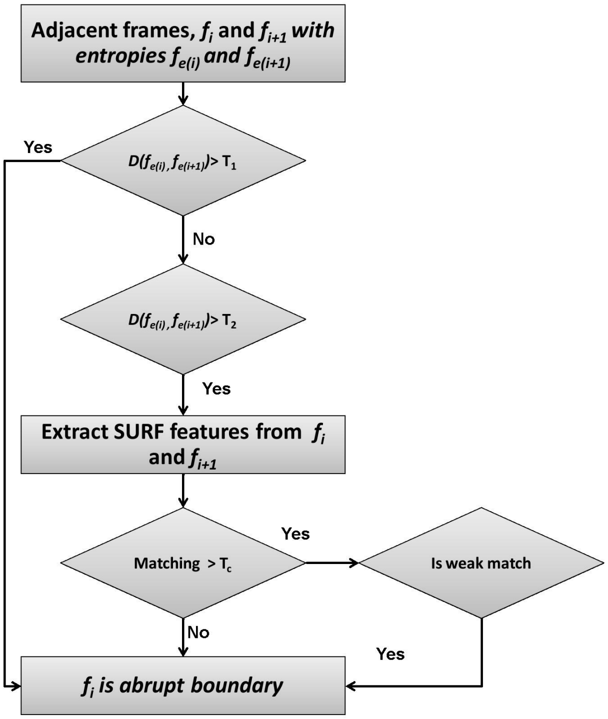

Adjacent frames, f_i and f_{i+1} with entropies $f_{e(i)}$ and $f_{e(i+1)}$
Yes
$D(f_{e(i)}, f_{e(i+1)}) > T_1$
No
$D(f_{e(i)}, f_{e(i+1)}) > T_2$
Yes
Extract SURF features from f_i and f_{i+1}
Yes
Matching $> T_c$
Is weak match
No
Yes
f_i is abrupt boundary
(a)

Fig. 6. Steps of abrupt shot detection process.

so that only those boundaries are detected that have obvious differences in visual contents, e.g., the examples shown in Figure 2. Since T_1 is high, it provides high precision but low recall. In a second step, to improve the recall, we use another lower threshold T_2 sufficient to detect all boundaries but with many false alarms. We treat these boundaries as candidate boundaries and attempt to eliminate the false alarms.

After detecting candidate boundaries, we use SURF keypoint correspondences to eliminate false alarms. If there are $C = \{c_1, c_2, \ldots, c_k\}$ candidate boundaries, where each c_i is an index of a frame in the video, then SURF features are extracted from f_{c_i} and f_{c_i+1} then matched using nearest neighbor search. In case of abrupt boundaries, f_{c_i} and f_{c_i+1} would have minimum feature matching. Let Q and R be the set of SURF features detected from f_{c_i} and f_{c_i+1}, respectively. We assume that point pair (q_i, r_j) is a match if the following conditions hold:

- The Euclidean distance d

$$d(q_i, r_j) = \min_{r_k \in R} d(q_i, r_k)$$

- and following inequality holds

$$d(q_i, r_j) < \min_{r_l \in R, l \neq j} d(q_i, r_l) \times \beta$$

where $0 < \beta < 1$, the smaller β , the fewer the matched points. We set β to 0.6. If the matching score between the

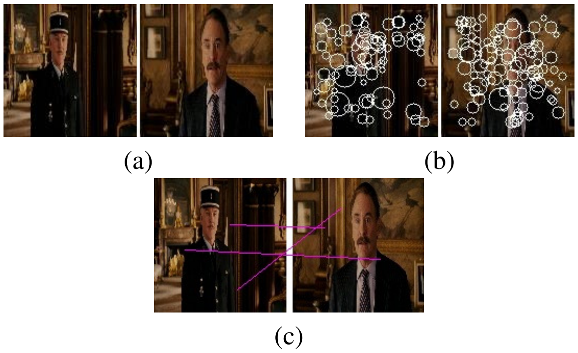


Fig. 7. Example SURF feature extraction and matching across a candidate shot boundary. a) Candidate boundary. b) SURF features. c) Matches between features.

features of f_{c_i} and f_{c_i+1} is less than T_c, then f_{c_i} is added to the set of shot boundaries S. We define a matching score as

$$\text{Matching score} = \frac{M}{|Q|} \times 100,$$

where M is the number of matched features and $|Q|$ is number of features in f_{c_i}. At the end of elimination process, S contains all the abrupt shot boundaries. Examples of feature extraction and matching are shown in Figure 7. Some boundaries eliminated due to the high matching between features are shown in Figure 8, where only the f_{c_i} and f_{c_i+1} frames are shown.

We find that during the false alarm elimination step, there are some abrupt boundaries that are missed due to weak matching (outliers) of features by nearest neighbor search. Examples are shown in Figure 9. To overcome weak matches, we compute the standard deviations of the keypoints' horizontal locations (σ_x) and vertical locations (σ_y) in each image. With a high capture rate, the shift in keypoint locations from frame to frame should not be larger than T_σ. Therefore, features can be restricted by using horizontal $\sigma_x^{f_{c_i}}$ and vertical $\sigma_y^{f_{c_i}}$ standard deviation. If horizontal and vertical standard deviation of SURF features in f_{c_i+1} frame is greater $\sigma_x^{f_{c_i}} + T_\sigma$ and $\sigma_y^{f_{c_i}} + T_\sigma$ then it is treated as weak matching and c_i is classified as abrupt boundary.

4. EXPERIMENTAL RESULTS AND DISCUSSION

We used the TRECVID-2007 shot boundary test set along with action movies, cartoons, and video lectures for testing the proposed shot detection method. For performance evaluation, we used the standard precision and recall measures:

$$\text{Recall} = \frac{N_{\text{boundaries detected}}}{N_{\text{true boundaries}}}$$

$$\text{Precision} = \frac{N_{\text{true boundaries detected}}}{N_{\text{boundaries detected}}}$$

Table 1. Threshold values for experiments

Threshold	Value	Description
T_f	fps $\times\alpha$	Frames per second with α, where α is number of seconds
α	0.5 to 2.0	Gradual effects can be detected in the interval from 0.5 to 2.0 seconds
T_1	20%	Percentage change in entropy from one frame to the next frame
T_2	2 %	
T_c	10%	Percentage of matching between SURF features
T_σ	9%	Threshold to control standard deviation

Table 2. Comparison of methods for shot boundary detection on TRECVID 2007.

	our framework	[9]	[10]
Precision	97.8%	96.99%	96.183%
Recall	99.3%	95.217	93.161

The values we used for the various thresholds are given in Table 1. We set them experimentally to obtain good performance.

On TRECVID-2007, the proposed method achieved the best published performance thus far. A comparison to existing results is shown in Table 2. Note that only 6% of the TRECVID-2007 boundaries are gradual. We obtain 93% precision and 100% recall over the fade boundaries. Our false alarm rate for for gradual shot boundaries is still relatively high due to increasing or decreasing entropy in dissolves and gradual appearance of objects in a scene.

We obtained good results for the abrupt shot boundary detection on videos and TRECVID-2007 test sets, we efficiently detected abrupt shot boundaries with 99.3% recall and 97.8% precision. Correspondence(matching) problem between local features is one of the key problems. We tackled this problem statistically using standard deviation. Results show that

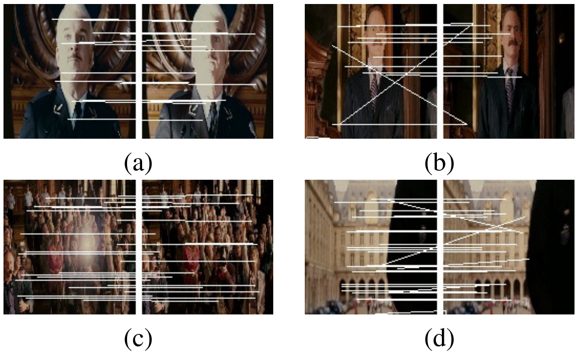
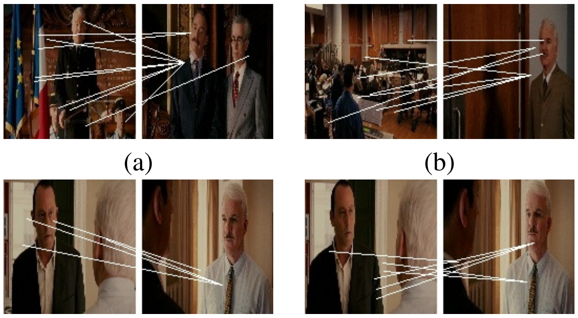


Fig. 8. False alarms eliminated using SURF features.



(a) (b)

(c) (d)

Fig. 9. Weak matching of SURF features

the features detected in f_i will not be scattered more than T_σ in f_{i+1} and this is due to the fact that there are more than 20 frames/second and very minor motion is also captured in hundred of frames. Standard deviation provided good results in videos but it is not efficient in images and object detection and recognition algorithms, e.g. in case of object detection, query image can be of various scales or may contain affine transformations.

SURF features are robust to various transformations but suffer in extreme low or high lighting in the movies such as in case of very bright or very dark images, the information of edges are lost. Therefore, SURF cannot find any features in the said scenario. SURF features also showed comparatively low performance in the presence of fade boundaries. To overcome these limitations, fade boundaries were detected and excluded prior to the application of SURF for abrupt shot boundary detection.

5. CONCLUSION

We have proposed a new method for shot boundary detection that is robust to camera motion, extreme illumination effects, object motion, and camera panning by making use of entropy difference analysis and SURF descriptor correspondences. The method produces good results on the TRECVID-2007 test set. We have also experimented with action movies, cartoons, dramas, and video lectures, with good results.

The proposed method could be used as a starting point for key frame extraction. Our preliminary experiments show that the median frame from each shot is a good representative of the entire shot. These key frames can be used for efficient video retrieval, video copy detection, and scene identification.

6. REFERENCES

[1] H. Zhang, A. Kankanhalli, and S.W. Smoliar, “Automatic partitioning of full-motion video,” *Multimedia Systems*, vol. 1, no. 1, pp. 10–28, 1993.

[2] Irena Koprinska and Sergio Carrato, “Temporal video segmentation: A survey,” *Signal Processing: Image Communication*, vol. 16, no. 5, pp. 477 – 500, 2001.

[3] Y. Tonomura, “Video handling based on structured information for hypermedia systems,” *Proceedings of ACM International Conference on Multimedia Information Systems*, pp. 333–344, 1991.

[4] A. Nagasaka and Y. Tanaka, “Automatic video indexing and full-video search for object appearances,” *Visual Database Systems II*, pp. 113–127, 1992.

[5] Z. Cernekova, I. Pitas, and C. Nikou, “Information theory-based shot cut/fade detection and video summarization,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 16, no. 1, pp. 82 – 91, jan. 2006.

[6] G. Camara Chavez, F. Precioso, M. Cord, S. Philipp-Foliguet, and Arnaldo de A. Araujo, “Shot boundary detection at trecvid 2006,” in *Proc. TREC Video Retrieval Eval.*, 2006.

[7] Xue Ling, Ouyang Yuanxin, Li Huan, and Xiong Zhang, “A method for fast shot boundary detection based on svm,” in *Image and Signal Processing, 2008. CISP ’08. Congress on*, May 2008, vol. 2, pp. 445 –449.

[8] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool, “Speeded-up robust features (surf),” *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, 2008.

[9] Jun Li, Youdong Ding, Yunyu Shi, and Wei Li, “Efficient shot boundary detection based on scale invariant features,” in *Image and Graphics, 2009. ICIG ’09. Fifth International Conference on*, 2009, pp. 952 –957.

[10] Y. Kawai, Sumiyoshi H., and N. Yagi, “Shot boundary detection at trecvid 2007,” *TRECVID 2007 Workshop*, 2007.