

Video Keyframe Analysis Using A Segment Based Statistical Metric in A Visually Sensitive Parametric Space

M. Omidyeganeh^{1'3}, S. Ghaemmaghami^{1'2}, S. Shirmohammadi³*

¹Electrical Engineering Department, Sharif University of Technology

² Electronics Research Center, Sharif University of Technology

³Distributed and Collaborative Virtual Environment Research Lab., University of Ottawa

m_omid@ee.sharif.edu, ghaemmag@sharif.edu, shervin@discover.uottawa.ca

Abstract

This paper addresses a new approach to the keyframe extraction problem employing Generalized Gaussian Density (GGD) parameters of wavelet transform subbands along with Kullback-Leibler distance (KLD) measurement. Shot and cluster boundaries are selected using KLDs between GGD feature vectors, and then keyframes are located based on similarity and dissimilarity criteria. Objective and subjective evaluations show the high accuracy of this new approach compared to traditional methods.

EDICS categories: TEC-MRS, ARS-SRE, SMR-SMD, SMR-REP.

Index Terms

Video keyframe extraction, generalized Gaussian density, Kullback-Leibler distance.

Authors addresses:

M. Omidyeganeh and S. Shirmohammadi: DISCOVER Laboratory, School of Information Technology and Engineering, University of Ottawa, 800 King Edward Ave., Canada, K1N 6N5 (Tel. +1 613 562-5800 extension 6206, Fax +1 613 562-5664)

S. Ghaemmaghami: Electronics Research Center, Sharif University of Technology, Azadi Ave, Tehran, Iran, P.O. Box 11155-8639 (Tel. +98 21 66005517, Fax +98 21 66030318)

1. INTRODUCTION

Keyframe extraction plays an important role in many video processing applications such as video compression, retrieval, skimming, editing, etc. Each video sequence is a combination of ‘shots’ captured from a certain viewpoint, with each shot consisting of a number of segments, each corresponding to a set of similar video frames. Keyframe extraction generally involves selecting one frame from each shot segment, called ‘cluster’, which represents that video segment. A keyframe should follow two main rules to adequately represent its cluster [1]: 1) it should be similar enough to the frames in its cluster; and, 2) it should tolerably differ from frames in other clusters. In a typical keyframe extraction process, first, a set of features are extracted from each frame to form a feature vector. Next, a suitable distance measure is applied to the feature vectors to examine similarity/dissimilarity between frames. Finally, based on the distance measurement in the selected feature space, shot and cluster boundaries are detected and the keyframes are extracted.

While research has been conducted to design efficient methods for keyframe selection, most of the works in the literature suffer from a variety of shortcomings that are addressed in section 2. In this paper, we propose a new parametric approach to the problem of keyframe extraction. In our method, features are selected based on the marginal statistical properties of 2D-wavelet transform subbands of each frame. These features represent the visual characteristics of the images; additionally, a suitable distance measure corresponding to our extracted features is employed to detect shot and shot cluster boundaries and select keyframes. The other characteristic of our work is that we have considered both the similarity and the dissimilarity rules. These two factors help us to choose the keyframes with higher accuracy.

The remainder of the paper is as follows. Related work is presented in section 2 and the GGD parametric model and the KLD metric are briefly described in section 3. In section 4, the proposed method and the measures used for keyframe extraction are introduced. The experimental results of shot and cluster boundary detection and keyframe extraction are given in section 5. The paper is concluded in section 6.

2. RELATED WORK

One of the simplest methods for keyframe extraction is to select the first frame of each shot [2, 3], but that may not be a proper choice, due to the fact that the selected frame does not necessarily represent the rest of the shot segment; also, most of the shots are composed of more than one cluster. Some other similar methods choose the central frame of the cluster or select keyframes by temporal uniform sampling of the signal. Yet other simple and fast techniques select the first frame of the shot as the keyframe and use the difference between the last defined keyframe and the current frame, and then select the frame as a new keyframe if the distance is larger than a given threshold. In such systems, however, the extracted keyframes do not necessarily convey the visual content of the video [2]. A more methodical technique is to select the keyframe based on minima in the motion or activity curves, like the approaches reported in [4, 5]; these methods, however, are of a high complexity and also may not select proper frames since the stillness is not a proper measure.

In another work, video sequences are modeled by an Auto-Regressive (AR) paradigm using color histograms [6], but the features used do not convey all contextual information of the frames, leading to lower accuracy in keyframe selection. An information-theoretic method is suggested in [7] to segment video sequences and extract the keyframes with a high recall rate (86.2%), which results in a poor accuracy rate (77.2%) in shot boundary detection. Clustering based approach is another solution introduced in [2], and also employed in [8], in which the nearest frames to the cluster centers are chosen as the keyframe, where they are of a high complexity. One of the main shortcomings of most of the mentioned approaches to keyframe selection is that they are only concerned with the difference between adjacent frames and systematically ignore the differences between frames over the video clusters. This is while cluster-based analysis could be quite more sensitive to group similarities/dissimilarities, compared to frame-based distance measurement that often deals with gradual changes without making any comparative, broad observations. This has been the first motivation behind our proposed approach and the reason we choose cluster-based analysis.

To realize the concept of cluster based analysis properly, there is a need to examine the visual contents of the video sequence for the keyframe selection. To achieve this, appropriate features, extracted from the video frames, are to be employed in the content analysis process. These features can

be motion information [4, 5], color histograms [2, 6, 9] or features extracted from a 2D-transform of the video frames [1], which have been used in existing keyframe selection methods. However, none of the existing methods are structurally matched with the Human Visual System (HVS), which we have found to be one of the main sources of the error in locating the keyframes. This has been the second major incentive behind our proposed approach; i.e, in our method, features are extracted from wavelet transformed subbands of each frame, leading to a better match with the HVS, since wavelet transform provides a sparse representation of signals and structurally conforms to the frequency sensitivity distribution of the HVS [10, 11].

2D wavelet based image modeling was first introduced in [12] and experimental results showed that the GGD function could yield a reasonable approximation to the marginal statistics of the 2D Wavelet domain coefficients of each subband using different filters [12, 13-15]. Together with the KLD, the GGD is used in [13] for texture retrieval in static images that led to excellent results. Motivated by the abovementioned ideas, we have utilized the GGD parameters to construct the feature vectors and used the KLDs between two GGD feature vectors to examine similarity between two frames in a cluster and discriminate between two frames from different clusters. To the best of our knowledge, no other work has introduced such an approach to video signals for the purpose of keyframe selection. The results are indeed more accurate and closer to the human selections in our subjective tests, compared to existing keyframe selection methods, as shown in section 5.

3. GGD PARAMETERS AND KLD DISTANCE MEASURE

The 2D wavelet based image modeling was introduced in [12] based on probability modeling. Minh N. Do. used the GGD features and Kullback-Leibler distances for texture retrieval and gained good results [13]. Accordingly, extracting two parameters of the GGD model, α and β , helps in estimating the density function of coefficients. The approximation of the probability distribution function (PDF) for the marginal density of a signal may be achieved by adaptively changing the two parameters α and β of the GGD [13, 16], defined as:

$$p(x, \alpha, \beta) = \frac{\beta}{2\alpha\Gamma(\frac{\beta}{\alpha})} e^{-\left(\frac{|x|}{\alpha}\right)^\beta} \quad (1)$$

where $\Gamma(\cdot)$ is the Gamma function, α models peak of the GGD and is called the scale parameter; and β is proportional to the inverse of the decreasing rate of the PDF and is the shape parameter. The PDF will be Laplacian if β is one and Gaussian when β is two. We calculate the GGD because, as mentioned earlier, investigations have shown that the GGD estimates give a good approximation to the 2D wavelet subband histograms [12, 15]. To measure the distance, D , the KLD is calculated between two GGD parameters of two equivalent wavelet subbands as [13]:

$$D(p(\cdot; \alpha_i, \beta_i) \parallel p(\cdot; \alpha_j, \beta_j)) = \log \left(\frac{\beta_i \alpha_j \Gamma(\frac{1}{\beta_j})}{\beta_j \alpha_i \Gamma(\frac{1}{\beta_i})} \right) + \left(\frac{\alpha_i}{\alpha_j} \right)^{\beta_i} \frac{\Gamma(\frac{\beta_i+1}{\beta_i})}{\Gamma(\frac{1}{\beta_i})} - \frac{1}{\beta_i} \quad (2)$$

Thus, taking into consideration the realistic assumption that coefficients in different subbands of one level are independent, the overall distance between two frames is the sum of the KLD between two equivalent subbands:

$$D(fv_i, fv_j) = \sum_{l=1}^{3L} D(p(\cdot; \alpha_i^{(l)}, \beta_i^{(l)}) \parallel p(\cdot; \alpha_j^{(l)}, \beta_j^{(l)})) \quad (3)$$

where, L is the number of transform levels, fv is the extracted feature vector explained in section 4, and i and j are the indices of the frames. At this stage, the KLD distances among the GGD feature vectors of each video frame are calculated and a look up table is constructed containing the KLDs between all frames of the video sequence.

4. KEYFRAME EXTRACTION METHOD

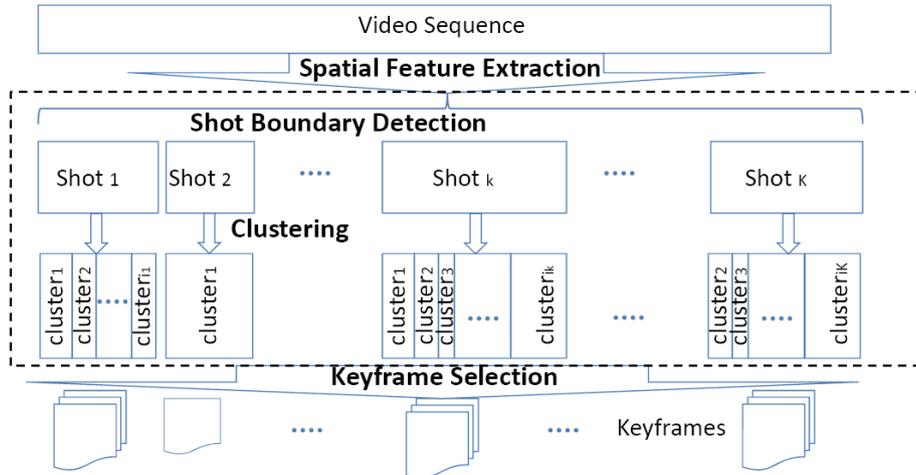


Figure 1. The proposed Keyframe selection system.

As mentioned in section 1, a keyframe should, in some sense, be similar to frames in its cluster and differ from other frames of the video sequence. As such, we need to consider both the similarity and the dissimilarity conditions. Here, we employ the KLD measure to fulfill this requirement. The proposed

algorithm for keyframe selection is carried out in three steps, as depicted in figure 1. First, the spectral parameters are extracted from each frame and the feature matrix F is constructed. Second, as the preprocessing step, the shot and cluster boundaries are detected. Finally, a keyframe is selected from each cluster regarding to the similarity and dissimilarity criteria. In the following 3 subsections, we explain each step in details.

4.1. Preliminaries

Assume that the video sequence, V , is composed of K shots and the k^{th} shot has i_k clusters, as:

$$V = \cup_{k=1}^K S_k \quad S_k = \cup_{j=1}^{i_k} C_{k,j}, \quad C_{k,j} = \cup_{n=1+b_{k,j}}^{b_{k,j}+n_{k,j}} f_n, \quad b_{k,j} = \sum_{k_1=1}^{k-1} \sum_{j_1=1}^{i_{k_1}} n_{k_1,j_1} + \sum_{j_1=1}^{j-1} n_{k,j_1} \quad (4)$$

where, S_k and f_n are pointers of the k^{th} shot and n^{th} frame, respectively. $C_{k,j}$ represents the j^{th} cluster of the k^{th} shot and contains $n_{k,j}$ frames, and $b_{k,j}$ is the number of preceding frames before the corresponding cluster. We want to extract a keyframe from each cluster.

4.2. Feature Extraction

Suppose that the video sequence contains N frames and we have extracted $2P$ GGD parameters from wavelet transform subbands of each frame as per equation (1); where $P = 3L$ and L is the number of wavelet transform levels. Thus, we will have a feature matrix F , the columns of which are extracted feature vectors and its size is $2P \times N$.

$$F = [f v_1 \ f v_2 \ \dots \ f v_n \ f v_N], \quad 1 \leq n \leq N$$

$$f v_n = [\alpha_{n,1} \ \beta_{n,1} \ \alpha_{n,2} \ \beta_{n,2} \ \dots \ \alpha_{n,p} \ \beta_{n,p} \ \dots \ \alpha_{n,P} \ \beta_{n,P}]^T, \quad 1 \leq p \leq P \quad (5)$$

4.3. Shot Boundary Detection and Clustering

In this step, the whole video stream is divided into K non-overlapping shots. To do so, the KLD between two adjacent video frames is calculated as given in (3) and a KLD vector is constructed. The KLD curve is sketched and both abrupt and gradual shot boundaries are selected based on a thresholding method defined experimentally. The threshold for shot boundary detection is calculated as $T_s = \rho m_w$, where m_w is the local mean of the KLD on a w size window which is chosen in the range of 4-10, and ρ is selected empirically as a number between 2 and 3.

Next, each shot is divided into one or more clusters. The KLD curve of each shot is smoothed and the local maximums of the smoothed KLD vector with values greater than a predetermined threshold are selected as cluster boundaries, here the threshold is chosen as $T_c = \rho m_{w+4}$.

4.4. Keyframe Selection

At this stage, first, the mean of the KLDs between the n^{th} frame of each cluster and the other frames in the cluster is calculated, using (3), as:

$$m_{dist}(n, C_{k,j}) = \frac{1}{n_{k,j}-1} \sum_{\substack{i \neq n \\ i \in C_{k,j}}} D(fv_n, fv_i) = \frac{1}{n_{k,j}-1} \sum_{\substack{i=1+b_{k,j} \\ i \neq n}}^{b_{k,j}+n_{k,j}} D(fv_n, fv_i) \quad (6)$$

This value shows the mean distance between frame n and other frames in its cluster. It is obvious from equation (6) that the $m_{dist}(n, C_{k,j})$ inversely relates to the similarity measure, so increasing the mean distance decreases the similarity. Next, we can calculate the mean distance of frame n in cluster $C_{k,j}$ and all the frames in other clusters as shown in (7) using equation (3):

$$m_{dist}(n, \bar{C}_{k,j}) = \frac{1}{N-n_{k,j}} \sum_{i \notin C_{k,j}} D(fv_n, fv_i) = \frac{1}{N-n_{k,j}} (\sum_{i=1}^{b_{k,j}} D(fv_n, fv_i) + \sum_{i=1+b_{k,j}+n_{k,j}}^N D(fv_n, fv_i)) \quad (7)$$

where $\bar{C}_{k,j}$ is the complementary part of the video sequence excluding $C_{k,j}$. In this case, the $m_{dist}(n, \bar{C}_{k,j})$ represents average of the distances between each frame and frames outside its cluster.

A keyframe should differ from the frames outside its cluster as much as possible, therefore $m_{dist}(n, \bar{C}_{k,j})$ has a direct relation with the dissimilarity measure; hence, increasing $m_{dist}(n, \bar{C}_{k,j})$ will also increase the discrimination factor. The final measure can be constructed by dividing, or subtracting, (6) by (7), as:

$$F_{k,j}(n) = \frac{m_{dist}(n, \bar{C}_{k,j})^x}{m_{dist}(n, C_{k,j})^y}, F_{k,j}(n) = x \cdot m_{dist}(n, \bar{C}_{k,j}) - y \cdot m_{dist}(n, C_{k,j}) \quad (8)$$

x and y are used to adjust similarity or dissimilarity measure in keyframe extraction, where they are both equal to one in our work. Thus, the keyframe of each cluster will be determined as:

$$keyf(k, j) = \operatorname{argmax}_{n \in C_{k,j}} F_{k,j}(n) \quad (9)$$

5. RESULTS AND DISCUSSION

We have used a number of natural video clips consisting of more than 20 hours video with different characteristics to evaluate our methods. Our test videos are selected from the ‘Hollywood-2 Human Actions and Scenes dataset (CVPR09)’ [17], the TRECVID 2006 shot boundary detection task data set

[18], the ‘Simon Fraser University (SFU) Video Library and Tools dataset’ [19], ‘The Open Video Project database’ [20], and other test videos. There are different spatial frame sizes in the video clips including QCIF, CIF and 16 CIF, and the sampling rates are 15, 24 or 29 frames per second. The above video repositories are widely used in the community and have sequences captured from different locations – indoor and outdoor - of different characteristics, and displaying human actions, car racing, news broadcasting, dogs running, airplane flying, glass breaking, etc. Also, they contain camera zooming, panning, translations and scene dissolves and fade in/out. Most of the test videos are very dynamic in both temporal and spatial domains, though there are also some static samples.

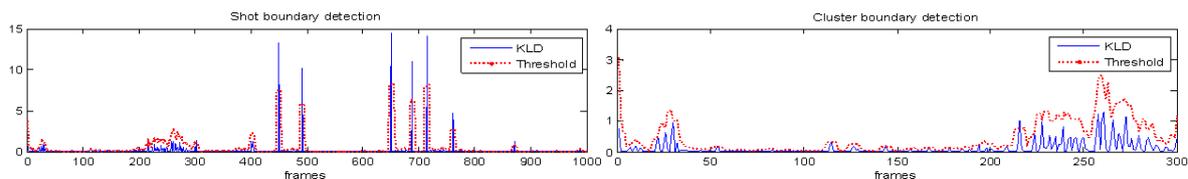


Figure 2. ‘sceneclipautoautotrain00060.avi’ shot boundaries: 303 450 492 651 689 715 762 871 989 (up), cluster boundaries: of the first shot: 54 115 144 194 209 (down).

We have applied 2D wavelet transform to all video frames, extracted the GGD parameters from each subband, and constructed the matrix F for each video. The video sequence is divided into shots based on the KLDs between two adjacent frames and a threshold calculated employing the mean KLDs of neighboring frames. Next, utilizing another threshold method explained in section 4.3, each video shot is decomposed into one or more clusters according to the local maximums of the smoothed KLD vector of the shot frames. One example of this procedure is shown in figure 2. The left plot shows the shot boundary detection where the dashed curve depicts the threshold in each frame and the blue curve is the KLD curve; and the right plot illustrates the clustering step.

Table 1. Details of video data set of the Shot Boundary Detection task in TRECVID 2006.

No. of Videos	Video Size	No. of frames	No. of Transitions	Transition Types			
				Cuts	Dissolves	Fade In/Out	Other
13	4.24 GB	597043	3785	1844(48.7%)	1509(39.9%)	51(1.3%)	381(10.1%)

To evaluate the proposed shot boundary detection algorithm, we have applied our algorithm to two famous video data sets: TRECVID 2006 and Hollywood2. TRECVID 2006 has 13 long representative news videos of 597,043 video frames in shot boundary detection. Table 1 contains some information about the transitions in this video set. Here the task is to locate the shot boundaries and define the type (cut or gradual) of the transition. First shot boundaries are selected based on the

threshold method described in section 4.2. Next, cut shots are detected among the selected shot boundaries. We expect to have a peak in the KLD curve at the cut shots where the curve decreases rapidly to reach zero, passing the transition frame. Suppose that frame i is selected as a shot boundary; the transition type is ‘cut’ if $|D(fv_{i+2}, fv_{i+1}) - D(fv_{i+1}, fv_i)| < t_p$ and $|D(fv_{i-1}, fv_{i-2}) - D(fv_{i-3}, fv_{i-2})| < t_p$ where the threshold t_p is chosen between .1 and .3 empirically. Figure 3 shows some examples of the KLD vector in different types of transitions. To find the location of the start and end frames of the gradual transitions, the start and the end points of increase and decrease of the KLD curve around the detected local maximum are selected.

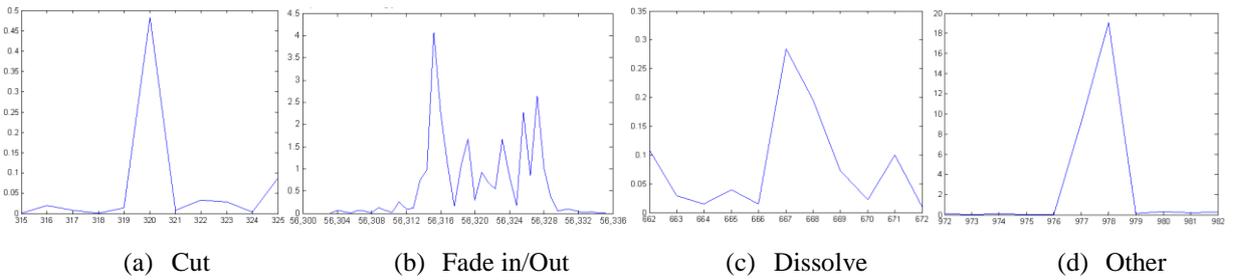
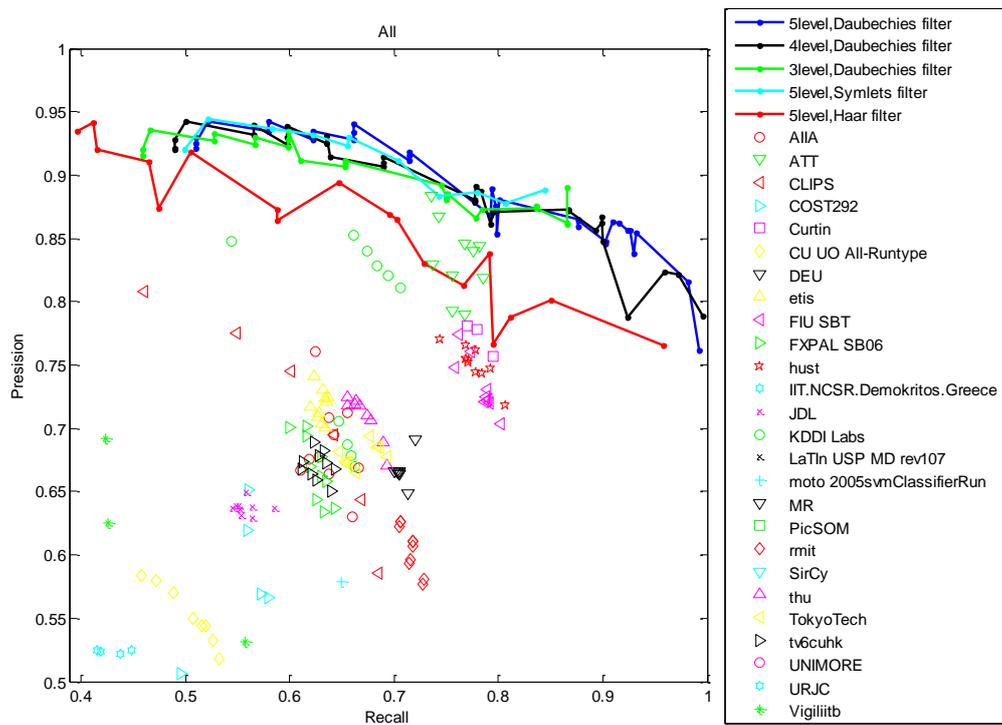


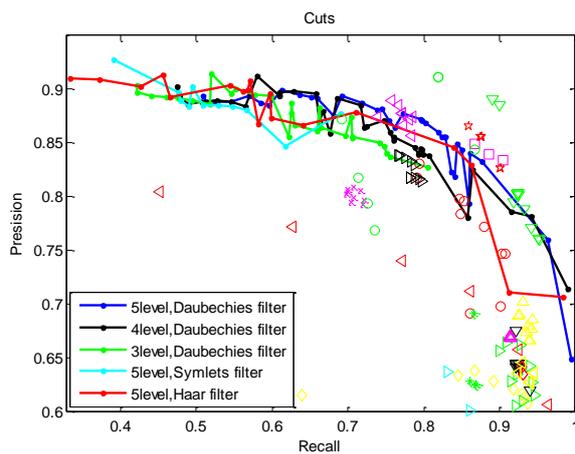
Figure 3. KLD curves around shot transitions for different types of transitions.

The performance of the algorithm is evaluated employing the recall and precision measures. For each wavelet filter – Haar, Daubechies and Symlets- and each transform level -3, 4 and 5 for Daubechies filter, threshold parameters ρ and w in shot boundary selection $T_s = \rho m_w$ and t_p in transition type detection- are varied, and recall and precision factors are calculated to form *precision to recall* curves. Thus the proposed method is compared with different methods in TRECVID 2006 Shot boundary detection task via these curves in figure 4. As is clear from the curves, our method well selects both cut and gradual shot boundaries. The high efficiency of the proposed method in gradual transition detection is also notable. This is important due to the well-known precision-sensitive problem [22, 23 and 24]: there are many visual events happening in natural video sequences (such as camera and object motions) which may be mistaken by a gradual transition –. But the precision of our system is robust against variations of parameters (figure 4c), where changing the recall rate from 0.3 to 1, the precision stays around 0.9. This shows the high performance of the method to avoid false detections in this kind of transition. The most important problem in discriminating the transition type occurs in gradual shot boundaries where the transition length is less than 4 frames. In other cases, the algorithm

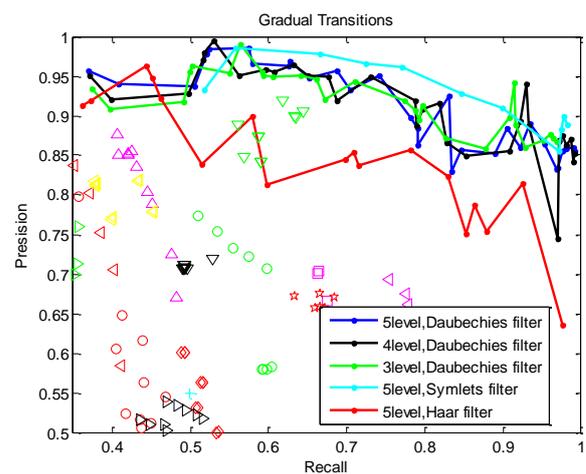
selects the shot boundaries with high performance. Increasing the wavelet transform levels, more delicate details are selected and the results improve.



(a) All transitions



(b) Cut transitions



(c) Gradual transitions

Figure 4. Precision to recall curves of shot boundary detection algorithm on TRECVID 2006 SB dataset.

The other dataset used here to evaluate the method is the Hollywood2 database. Hollywood2 contains 1152 video clips collected from more than 69 movies consisting of 1,025,278 video frames containing 8199 shot transitions including cuts and gradual transitions. Figure 5 shows an example of the KLD vector versus video frames for our method and the distance curve proposed in [6] for shot boundary

detection. The test video is ‘sceneclipautoautotrain00077.avi’ from the Hollywood2 data set. The plot confirms the accuracy of our method, where the shot boundaries in frames 358, 390, and 424 are not detected by the method in [6], because it is only based on the color features and does not detect scene changes when the color distributions of both scenes are not too different. Two examples of shot boundaries are shown in figure 6, where the method [6] has not detected the scene change in the second example, but our method has detected both shot boundaries. We have calculated precision to recall values for the shot boundary detection results varying the threshold parameters, ρ and w , for different wavelet filters and numbers of transformed levels. Figure 7 compares these curves with the ones obtained from [6], changing the threshold and AR order parameters. The results confirm the improvements achieved by our proposed method.

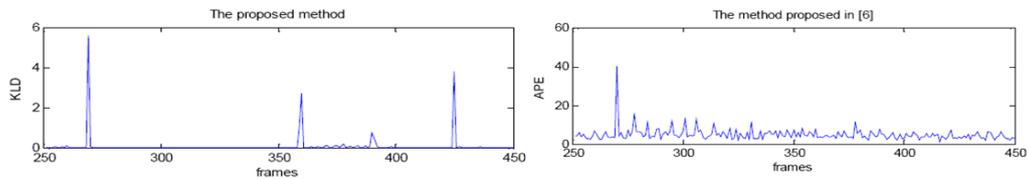


Figure 5. Preprocessing stage. Shot boundaries located at frames 268, 358, 390 and 424, using our proposed method (left) and the method in [6] (right).



Figure 6. Some examples of shot boundaries.

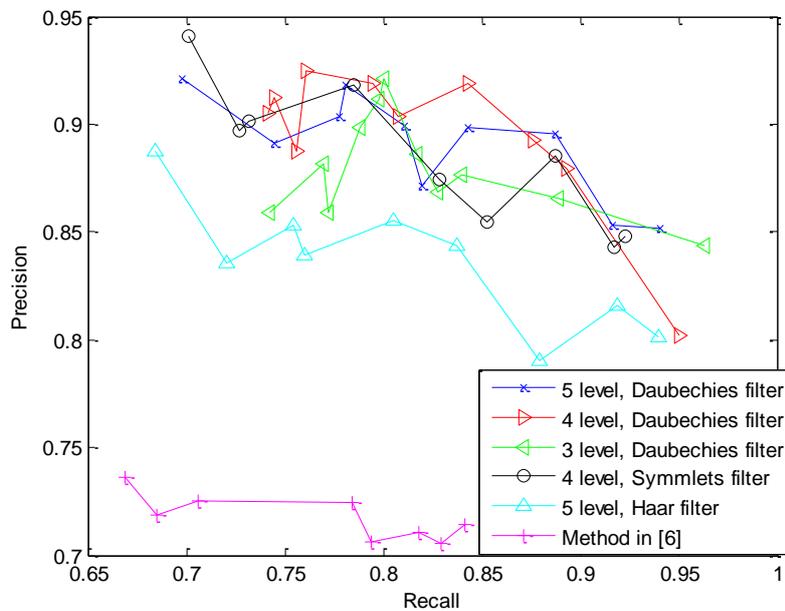


Figure 7. Precision to recall curves of shot boundary detection algorithm on Hollywood 2 Database.

The drawback of our proposed method is with heavily compressed videos, where the blocking effect is obvious. The reason is the wrong information received from wavelet based features relying on the edges. The video ‘sceneclipautoautotrain00060.avi’ from Hollywood2 dataset is an example. Using color features along with wavelet based features will lead to more reliable results than our method.

In the next step, using a manual threshold, each shot is divided into one or more clusters. Then, keyframes are extracted from the video sequence using equation (10) where one keyframe is selected from each cluster. We have applied the shot clustering and keyframe extraction idea to all of the videos in the data set. Figure 8 illustrates an example of the keyframe extraction. Three keyframe selection measures are applied to video segments. As shown in the figure, some of the keyframes extracted by ignoring the dissimilarity measure are still too similar (e.g., the second and the third keyframes in figure 8.b). This is while the other two measures select distinct keyframes properly. Also, comparing the subtractive measure (8.a) with the rational one (8.c), the former often selects better results.

In the keyframe extraction step, the proposed method again outperforms the color histogram method, since it can recognize details that may not be conveyed by the color statistics. An example is shown in figure 9. The color histogram scheme has chosen the incorrect frame, while our method decides the proper frame as the keyframe because of its ability to locate edges and textures in the image, as well as details that are contextually more important than colors. We can see in figure 9.c, where the keyframe extraction results of method [6] are depicted, that some of the frames have no representative keyframe; for example the frames in the fourth row of figure 9.a. But in the case of wavelet based features, a more accurate keyframe selection is achieved, because the wavelet space comes with a better approximation to the HVS. Furthermore, we have compared our features with the DCT features employed in [1] and an example is shown in figure 9.d. The HVS is too sensitive to edges and textures which cannot be captured by low frequency coefficients of the DCT transform, while the wavelet transform well captures these changes. As shown, no keyframe is selected from among the frames in the fourth row of figure 9.a by the method in [1].

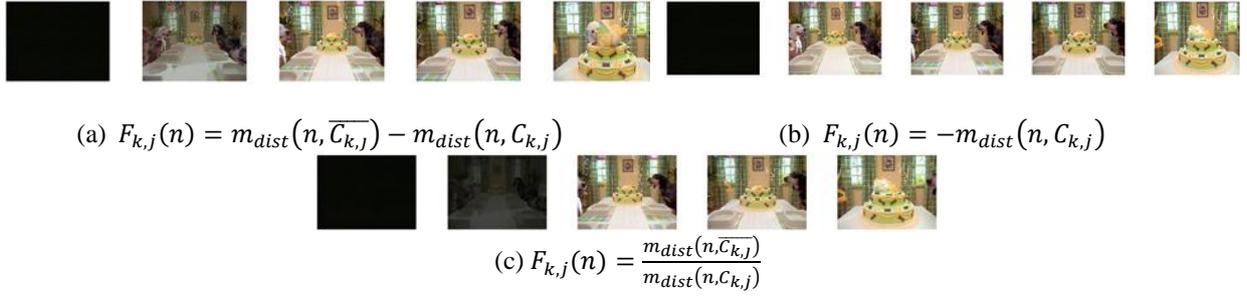


Figure 8. Keyframe extraction results: 5 shots, 9 clusters. GGD features of 4-level wavelet transform with filter ‘Daubechies’. KLD measure.

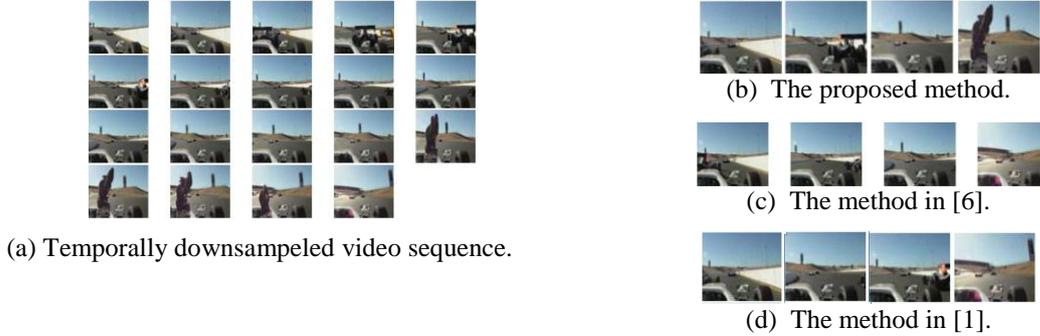


Figure 9. Comparison of keyframe methods, GGD features of 4-level wavelet transform with filter ‘Daubechies’. KLD measure.

Since there is no standard human-in-the-loop method for evaluation of the keyframe extraction results, in addition to the above objective comparisons, we have also used a subjective test to evaluate our method [21]. We asked 11 subjects who have some knowledge about video processing methods to rate the results of our keyframe extraction method and the method in [6] as correct or false. The precision to recall curve is sketched in figure 10 for different tests employing different system parameter. Subjective test results demonstrate the efficiency of our method

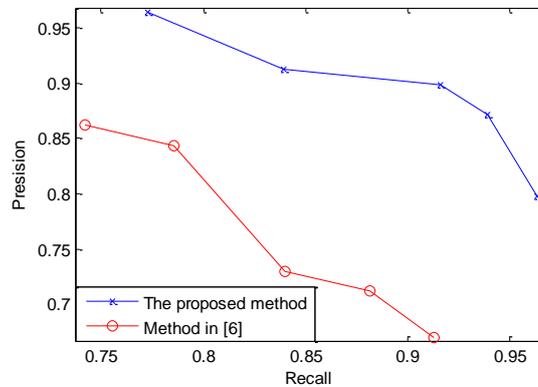


Figure 10. Subjective evaluation of the proposed method, #frames: 67,200, #extracted keyframes: 589.

6. CONCLUSIONS

We have investigated the keyframe extraction problem from a statistical modeling perspective by employing GDD for feature selection and KLD as a similarity measure. The GGD parameters, extracted from the wavelet transform subbands of each frame, are used to construct the frame feature vector, where the KLD represents the distance between the GGD feature vectors. In the first step, the KLDs between adjacent frames are utilized to determine shot boundaries and segment shots into clusters. Next, the keyframes are properly selected based on both similarity and dissimilarity criteria, within and outside the corresponding cluster, respectively. Experimental results confirm improvements achieved by our proposed method over existing shot boundary detection and keyframe extraction algorithms. This improvement is achieved due to the conformity of the 2D wavelet structure to the HVS characteristics and, specifically, because of higher sensitivity of the KLD, in the selected feature space, to the scene change. This keyframe extraction method can be used in different video applications, including scene search/retrieval, content based summarization, and video editing.

REFERENCES

- [1] M.L. Cooper and J. Foote, "Discriminative techniques for keyframe selection," *Proceedings of the Int'l Conference on Multimedia and Expo, ICME*, pp. 502-505, 2005.
- [2] Y. Zhuang, Y. Rui, T. S. Huang and S. Mehrotra, "Adaptive key frame extraction using unsupervised clustering," *IEEE Int'l Conf. on Image Processing*, pp. 283-287, 1998.
- [3] A. Nagasaka and Y. Tanaka, "Automatic video indexing and full-video search for object appearances," *Visual Database Systems II, Elsevier*, pp. 113-127, 1992.
- [4] E. Bulut and T. Capin, "Key Frame Extraction from Motion Capture Data by Curve Saliency," *Proceedings of CASA 2007-Computer Animation and Social Agents*, June 2007.
- [5] L. Shao and L. Ji, "Motion Histogram Analysis Based Key Frame Extraction for Human Action/Activity Representation," *6th Canadian Conference on Computer and Robot Vision (CRV)*, pp. 88 – 92, 2009.
- [6] W. Chen and Y.J. Zhang, "Parametric model for video content analysis," *Elsevier B.V., Pattern Recognition Letters*, vol. 29, pp. 181–191, 2008.
- [7] B. Janvier, E. Bruno, T. Pun and S.M. Maillet, "Information-theoretic temporal segmentation of video and applications: multiscale keyframes selection and shot boundaries detection," *Multimedia tools and application*, vol.3, no. 3, pp. 273-288, Sept. 2006.
- [8] W.L. Zhao, C.W. Ngo, H.K. Tan, X. Wu, "Near-Duplicate Keyframe Identification With Interest Point Matching and Pattern Learning," *Multimedia*, vol. 9, no. 5, pp. 1037-1048, 2007.
- [9] Z. Sun, K. Jia, H. Chen, "Video Key Frame Extraction Based on Spatial-Temporal Color Distribution," *IHMSP*, pp.196-199, 2008.
- [10] S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Trans. Patt. Recog. and Mach. Intell.*, vol. 11, no. 7, pp. 674– 693, July 1989.
- [11] T.H. Oh and R. Besar, " JPEG2000 and JPEG: image quality measures of compressed medical images," *Telecommunication Technology, NCTT Proceedings. 4th National Conference on*, pp. 31 – 35, Jan. 2003.

- [12] E.P. Simoncelli and R.W. Duccigrossi, "Embedded Wavelet Image Compression Based on a Joint Property Model," *4th IEEE Int'l Conf. Image Processing, VoU*, pp. 640-643, 1997.
- [13] M.N. Do, "Directional Multiresolution Image Representations," PhD thesis, Swiss Federal Institute of Technology, 2001.
- [14] P. Moulin and J. Liu, "Analysis of multiresolution image denoising schemes using generalized Gaussian and complexity priors," *IEEE Trans. Inform. Th.*, vol. 45, no. 3, pp. 909-919, 1999.
- [15] K. Sharifi and A. Leon-Garcia, "Estimation of shape parameter for generalized Gaussian distributions in subband decompositions of video," *IEEE Trans. Circuits Sys. Video Tech.*, vol. 5, no.1, pp. 52-56, Feb. 1995.
- [16] D.Y. Duncan and M.N. Do, "Directional Multiscale Statistical modeling of Images," *Proc. SPIE The Int'l Society for Optical Engineering*, vol. 5, 207, pp. 69-79, 2003.
- [17] <http://www.irisa.fr/vista/Equipe/People/Laptev/download.html>.
- [18] <http://www-nlpir.nist.gov/projects/trecvid>, National Institute of Standards and Technology (NIST).
- [19] http://nsl.cs.sfu.ca/wiki/index.php/Video_Library_and_Tools.
- [20] <http://www.open-video.org>.
- [21] M.J. Pickering and S. Ryger, "Evaluation of key frame-based retrieval techniques for video," *Elsevier Computer Vision and Image Understanding, CVIU*, vol. 92, no. 2-3, pp. 217-235, 2003.
- [22] C. Cotsaces, N. Nikolaidis and I. Pitas, "Video shot detection and condensed representation: a review," *IEEE Signal Processing Magazine*, vol. 23, no. 2, pp. 28-37, 2006.
- [23] A. Amiri and M. Fathy, "Video Shot Boundary Detection Using QR-Decomposition and Gaussian Transition Detection," *EURASIP Journal on Advanced Signal Processing*, 2009.
- [24] A.F. Smeaton, P. Over and A.R. Doherty, "Video shot boundary detection: Seven years of TRECVID activity," *Computer Vision and Image Understanding*, pp. 411-418, 2010.