

# GROUP BASED SPATIO-TEMPORAL VIDEO ANALYSIS AND ABSTRACTION USING WAVELET PARAMETERS

*M. Omidyeganeh\*<sup>1,3</sup>, S. Ghaemmaghami<sup>1,2</sup>, S. Shirmohammadi<sup>3</sup>*

<sup>1</sup>Electrical Engineering Department, Sharif University of Technology

<sup>2</sup> Electronics Research Institute, Sharif University of Technology

<sup>3</sup>Distributed and Collaborative Virtual Environment Research Lab., University of Ottawa

[m\\_omid@ee.sharif.edu](mailto:m_omid@ee.sharif.edu), [ghaemmag@sharif.edu](mailto:ghaemmag@sharif.edu), [shervin@discover.uottawa.ca](mailto:shervin@discover.uottawa.ca)

## Abstract

In this paper, we present a spatio-temporal event based approach to video signal analysis and abstraction employing wavelet transform features. The video signal is assumed to be a sequence of overlapping independent visual components called events, which typically are temporally overlapping compact functions that describe temporal evolution of a given set of the spatial parameters of the video signal. We utilize event based temporal decomposition technique to resolve the overlapping arrangement of the video signal that is known to be one of the main concerns in video analysis via conventional frame based schemes. In our method, a set of spatial parameters, extracted from the video, is expressed as a linear combination of a set of temporally overlapping compact functions, called events, through an optimization process. First, to reduce computational complexity, the video sequence is divided into overlapped groups. Next, Generalized Gaussian Density (GGD) parameters, extracted from 2D wavelet transform subbands, are used as the spatial parameters. Temporal Decomposition (TD) is then applied to the GGD parameters, structured as a frame based matrix of GGD vectors, to compute the event functions and associated orthogonal GGD parameters. Frames located at event centroids, which are much smaller in number than the number of frames in the original video, are taken as candidates for the

---

### 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 Institute, Sharif University of Technology, Azadi Ave, Tehran, Iran, P.O. Box 11155-8639 (Tel. +98 21 66005517, Fax +98 21 66030318)

keyframes that are selected based on a distance criterion in the feature space. Our contribution is that this still image video abstraction scheme does not need shot or cluster boundary detection, unlike current methods. Experimental results confirm the efficiency and accuracy of our approach.

**Keywords**— Video abstraction; visual events; 2D wavelet features; video temporal decomposition.

## 1. INTRODUCTION

Video abstraction is a process to produce a summary of the content of the video sequence [20, 21]. This summary could give key information about the video to the user in a much shorter time than the length of the original signal. There are two major categories of video abstracts: still image abstracts –keyframes - and moving image abstracts – video skims. Still image abstraction is much faster than video skim abstraction, since there is no need to consider the audio and textual information and timing or synchronization issues of the video; rather, only the visual content of the signal is required. Also, by displaying the keyframes based on the temporal order of the extraction process, users can realize the visual content more easily and rapidly. On the other hand, video skims may contain more information and could be more fascinating to users [21]. In this work, we propose a method for real-time still image video abstraction using a group based analysis scheme.

Keyframes convey the main information about contents of a video stream and represent the video by a number of representative frames. Keyframes can be employed in several video processing applications, e.g. scene detection, summarization, retrieval, abstraction, editing, and compression. In traditional keyframe extraction methods, some features are chosen from each frame, and then shot and cluster boundaries, and subsequently keyframes, are selected based on a distance measure in the feature space.

In this paper, we propose a novel fast method for video abstraction that is based on a local correlation analysis within a group of successive frames to locate the visual events over the video stream. This is achieved by employing Temporal Decomposition (TD) that is applied to a given set of features, extracted from a segment of the video signal, to

determine the local correlations through an orthogonalization procedure in the feature space. In our approach, first, the video sequence is divided into overlapped groups of frames, each composed of 200-300 frames. This group based approach makes it possible to apply the abstraction to long videos while the complexity remains low and the algorithm is fast. Next, TD is applied to each group to extract the candidate frames for the keyframe selection process. This preselecting procedure replaces the currently common process of cluster boundaries detection, which is incapable of resolving the overlapping nature of visual events. Our method selects appropriate spatial parameters from each frame based on marginal statistical properties of 2D-wavelet transform subbands, creating a parametric space based on the characteristics of Human Visual System (HVS). To extract features from frames, we utilize wavelet transform which has been shown to give a sparse representation of the image signal and matches with frequency distribution function of the HVS [9-12]. In the next step, the keyframes are selected from among the visual event centroids of each group based on the Kullback-Leibler Distance (KLD) between the corresponding Generalized Gaussian Density (GGD) parameters. These spatial parameters along with the TD and grouping technique as well as the suitable distance measure – KLD - help reach the high efficiency of our method. Also, the employed group based analysis transfers the signal into a compact sequence of keyframes without missing key visual information about the signal. So, any further content-based analysis can be applied to the compact signal resulting from this process.

The paper is structured as follows. Related work is presented in Section 2 and a brief description of video temporal evolution approximation is given in Section 3. The proposed algorithm for video abstraction is presented in Section 4. Section 5 contains the experimental results and Section 6 concludes the paper.

## **2. RELATED WORK**

One of the simplest approaches to keyframe selection is to choose the first or the middle frame of each shot [1, 2] as the keyframe. However, this does not necessarily result in a

suitable selection. Furthermore, often more than one keyframe is needed to describe a shot. However, other quick methods choose the first frame of the shot as the keyframe and employ the dissimilarity between the last selected keyframe and the recent frame. A frame is then picked as a new keyframe if the distance exceeds a certain threshold, though the selected keyframes may not express well the visual content of the video [1]. Keyframe extraction, using the minima in the motion or activity curves, is a more systematic procedure [4, 5], where it could be computationally expensive and may not extract apt frames, because stillness is not an appropriate measure.

In [4], Auto-Regressive (AR) modeling is used to model a video signal based on color features. However, the features employed do not properly express visual and textual information of the signal, causing low accuracy in the keyframe selection process. An information-theoretic scheme is proposed in [22] that segments video sequences and extracts the keyframes with a high recall rate and poor accuracy rate in shot boundary detection. Clustering based approach is a computationally complex solution introduced in [1], in which the nearest frames to the cluster centers are chosen as the keyframes. To examine the visual contents of the video sequence for keyframe selection, appropriate features should be extracted from the video frames. These features can be motion information [3, 24], color histograms [1,4] or features extracted from a 2D-transform of the video frames [23], which have been used in existing keyframe selection methods.

While the above approaches lead to a certain level of success in keyframe selection, they suffer from three following major shortcomings which are addressed in this work:

- 1) They mostly need a video shot/cluster boundary detection stage prior to the keyframe selection process [1-5, 22-23]. This stage become time consuming and increases the computational load of the abstraction algorithm.
- 2) Most current solutions to the problem rely only on the differences between adjacent frames that may not properly discriminate the keyframes from other frames [1-5, 22-23]. This is likely to happen when the visual contents of adjacent frames change quite slowly.

- 3) Some of the previous works employ inappropriate features which cause the selection of improper keyframes [1, 4 and 22].

These three shortcomings are addressed in our proposed method, in which no shot or cluster segmentation is required. Furthermore, unlike existing methods, our proposed method determines the differences between all candidate frames to capture both rapid and slow evolution of frames and selects more appropriate keyframes comparing the condition when only differences between adjacent frames are used. Finally, proper features based on Human Visual System (HVS) characteristics are utilized to improve the results. We show that this selection method significantly improves the performance of the keyframe extraction process, as compared to the conventional methods.

### **3. APPROXIMATION OF VIDEO TEMPORAL EVOLUTION**

Before presenting our algorithm in section 4, in this section we discuss the approximation and modeling background used in our algorithm. The video signal can be assumed to be a succession of individual *visual events* emerging successively with/without overlap with their neighbors. By *visual event*, we mean any specific visual activity that extends over time. For example, a car passing the road, a cat jumping on top of a table, or a man entering the office. Thus the video sequence can be considered as a combination of the consequent visual events, which may appear separately, simultaneously, or with overlap in time. Video events are:

- 1) Different kinds of shot changes (fade in/out, cuts) and camera zooming, padding, and object emergence – which cause shot clustering.
- 2) Temporal objects extended over segments of the signal that are often classified into three main categories: activities, motions, and temporal textures [5,6].

Video events can be characterized by some parameters:

- 1) Duration over time, which varies from small – object emergence - to large – news man speaking over several frames - number of frames.

- 2) Possible overlap with adjacent events, for example zooming while an object emergence occurs.

Each event is a temporal visual activity expanding over the time domain, where the temporal duration and pattern could differ for different events. Neighboring events may overlap in time; for example, during zooming, an object emergence may happen or there could be a temporal overlap between two video shots in fade in/out shot transitions. This is while each visual event can be assumed to be independent of other visual events, considering the reasonable assumption that the video director has already thought about the occurrence of each of them individually. This overlapping structure makes the analysis of the video signal highly complicated, because there may be no apparent boundaries between individual visual activities. However, we show that TD is able to describe such complicated relations using an appropriate set of spatial parameters.

TD was originally introduced in [7] to extract overlapping phonetic events of speech using an apt set of spectral parameters. Such a decomposition method was later shown to be an effective approach to audio processing, by locating audio events [8]. The capability of resolving the temporal overlap between adjacent events can be employed in video analysis, as we have shown in this work. TD represents feature vectors of a video block by a weighted sum of event functions – each corresponding to a visual event - and models the temporal evolution of the feature vectors, in a given signal block, by separately analyzing the contribution of successive basis functions to the statistical characteristics of the feature parameters. This analysis procedure results in an orthogonal matrix of feature vectors, called *target* vectors, and a sparse matrix of basis functions, called *events* or *targets*. The method can be formulated as:

$$Y = A\Phi \tag{1}$$

where  $Y$  is the matrix of parameters of size  $q \times N$ ,  $N$  is the number of frames and  $q$  is the total number of parameters extracted from each frame.  $\Phi$  is the matrix of event functions

of size  $m \times N$ ,  $A$  is the matrix of target vectors with size  $q \times m$ , and  $m$  is the number of event functions in the interval  $n=1$  to  $n=N$ .

Equation (1) can be written in scalar form as a set of linear equations, where each equation describes the time evolution of the  $i^{\text{th}}$  parameter of  $Y$ 's columns:

$$\hat{y}_i(n) = \sum_{k=1}^m a_{ik} \phi_k(n), \quad 1 \leq n \leq N, 1 \leq i \leq q \quad (2)$$

where,  $\hat{y}_i(n)$  is the  $i^{\text{th}}$  parameter of frame  $n$  approximated by the model,  $\phi_k(n)$  is the  $k^{\text{th}}$  event function at frame  $n$ , and  $a_{ik}$  is the weighting factor. In these equations, only the matrix  $Y$  is known and we need to decompose  $Y$  through orthogonalization to compute  $A$  and  $\Phi$  matrices [8].

The process of TD is mainly completed in two phases. First, the matrix  $Y$  is divided into overlapping sub-blocks, with predefined length  $l_{sb}$ . Then, the positions of event functions of each sub-block are identified by a Singular Value Decomposition (SVD) of the spectral parameters matrix of each sub-block; this division makes the SVD process much faster than applying the SVD to the whole block. To complete this step, event functions are refined iteratively to minimize the distance between the estimated and the original parameter sets, and the algorithm selects the best event functions to describe the temporal evolution of the consequent spatial features.

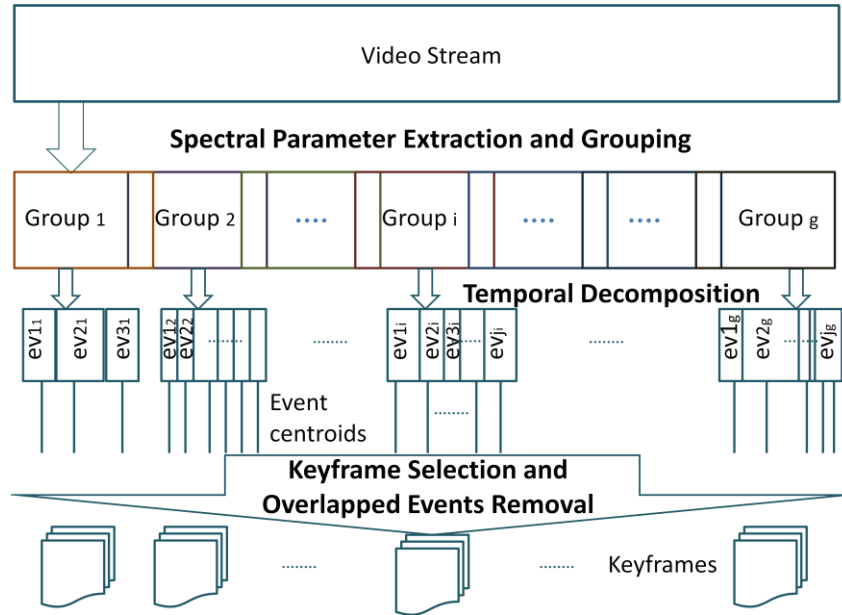
The length of each sub-block,  $l_{sb}$ , is in fact the temporal resolution of the TD process. Decreasing this value reduces the probability of missing short visual events and increases the probability of splitting a long event into some short visual events. We have selected this value experimentally around the video temporal sampling rate to cover all visual events lasting around a second and keeping the computational complexity low. After selecting the events from each sub-block, the optimal events and their locations are selected over the whole video group based on the negative-going zero crossings of the timing function  $v_k(n_{c_k})$ , given in [7] as:

$$v_k(n_{c_k}) = \frac{\sum_n (n - n_{c_k}) \phi_k^2(n)}{\sum_n \phi_k^2(n)} \quad 1 \leq k \leq m \quad (3)$$

where  $n_{c_k}$  is the index of the  $k^{th}$  event centroid of the  $k^{th}$  event function, the sum is calculated over the whole video group, and  $m$  is the number of events.

The first phase is performed through the above-mentioned analysis procedure applied to each sub-block of a given block of the video signal to locate event centroids. There is a refinement step employed in Temporal Decomposition methods [7,8] to reduce the decomposition error and has no effect on the location of the event cancroids. In our method we have no need for such step and have therefore not used it.

#### 4. PROPOSED ALGORITHM



**Figure 1.** The proposed keyframe extraction method.

As mentioned in previous sections, the proposed method is different from current schemes and seeks for keyframes within the signal through a near-optimal process that is highly sensitive to the temporal correlation between all frames within each video group. The other advantage of our method is that it employs group-based TD analysis of spatial features which successfully resolves the overlapping nature of the visual evolution of long video signals and makes it possible to be applied in real-time applications by dividing the video sequence into overlapping groups through a simple stage. We decompose each group of video frames into visually overlapping events by searching for

local correlations in the feature space that is generally based on the frame contents. By utilizing this analysis technique, only a small fraction of the video sequence, just the frames located at the most steady instants of the signal, are utilized in the keyframe extraction process.

In this section, the scheme to extract keyframes is introduced. There are four steps to select the keyframes, as shown in figure 1. First, spatial features are extracted from video frames and the matrix of parameters is constructed. Second, the video signal is divided into identically overlapping video groups, where each group is a combination of visual events. It should be noted that the first and the second steps can be swapped and the results will be the same in both cases. Next, utilizing video temporal decomposition, event centroids of each group of frames are selected. Finally, overlapping events are removed and keyframes are selected from among event centroids based on a certain criterion. In the following four subsections, each step of the procedure is described in detail. After presenting our proposed approach, we proceed with a discussion of the computational complexity of our approach in section 4.5.

#### 4.1. Spatial Feature Extraction

We use the 2D wavelet parameters as the spatial parameters which convey visual characteristics of the frames. The Human Visual System (HVS) is too sensitive to edges and textures. The wavelet transform structurally matches with the HVS characteristics and perfectly captures these variations [9,10]. Thus, we use wavelet based parameters to construct the feature vectors. It has been shown that the GGD function well estimates the distribution of the 2D wavelet transformed coefficients of each subband, employing different filters and transform levels [9,11 and 12]. The GGD function is defined as:

$$p(x, \alpha, \beta) = \frac{\beta}{2\alpha\Gamma(\frac{1}{\beta})} e^{-\left(\frac{|x|}{\alpha}\right)^\beta} \quad (4)$$

where  $\Gamma(\cdot)$  is the Gamma function,  $\alpha$  is the scale parameter that forms the width of the function, and  $\beta$  is the shape parameter that increases inversely with decreasing rate of the

probability distribution function (PDF). The PDF is Laplacian if  $\beta = 1$  and Gaussian if  $\beta = 2$ . Thus, extracting two GGD parameters ( $\alpha$  and  $\beta$ ) will give sufficient information to find out the marginal histograms.

KLD, also known as information divergence, shows the difference between two probability distributions. This asymmetric non-negative measure, between two distributions  $P$  and  $Q$ , is defined as:

$$D(P\|Q) = \sum_x p(x) \log \left( \frac{p(x)}{q(x)} \right) \quad (5)$$

In this case after some operations, the KLD between two GGD parameters of two equivalent wavelet subbands is employed and calculated as [12]:

$$D(p(\cdot; \alpha_i, \beta_i) \parallel p(\cdot; \alpha_j, \beta_j)) = \log \left( \frac{\beta_i \alpha_j \Gamma\left(\frac{1}{\beta_i}\right)}{\beta_j \alpha_i \Gamma\left(\frac{1}{\beta_j}\right)} \right) + \left( \frac{\alpha_i}{\alpha_j} \right)^{\beta_i} \frac{\Gamma\left(\frac{\beta_j+1}{\beta_i}\right)}{\Gamma\left(\frac{1}{\beta_i}\right)} - \frac{1}{\beta_i} \quad (6)$$

Consequently, with the realistic assumption of independence of the different subbands of the same level, the distance between two GGD feature vectors of two frames is the summation of the KLDs between two corresponding subbands.

Using the above parameters extracted from the video signal, we define the parameter matrix  $Y$ , as:

$$Y = [fv_1 \ fv_2 \ \dots \ fv_i \ fv_N], \quad fv_i = (fv_{GGDi})^T \quad (7)$$

Thus, the distance between two frames  $i$  and  $j$  is defined as:

$$\text{dist}(i, j) = \text{dist}_{\text{KLD}}(fv_i, fv_j) = D(fv_i, fv_j) = \sum_{l=1}^{3L} D(p(\cdot; \alpha_i^{(l)}, \beta_i^{(l)}) \parallel p(\cdot; \alpha_j^{(l)}, \beta_j^{(l)})) \quad (8)$$

where  $L$  is the number of transform levels,  $fv$  is the extracted feature vector,  $i$  and  $j$  are the indices of the frames, and the subband LL is excluded.

## 4.2. Grouping

To extend the video temporal decomposition idea as a spatio-temporal video analysis approach suitable for various video applications, we have proposed group based temporal decomposition. The video sequence is divided into a sequence of equal-length overlapping groups of frames and the temporal decomposition is applied to each group. The overlap between adjacent groups prevents missing the visual events in the group boundaries. The size of each video group,  $N_g$ , is taken in the range of 200-300 frames based on the TD mathematical characteristics to prevent increasing the computational time. Also, the overlap between groups,  $N_{ov}$ , is chosen within 15-25 frames based on event function durations to avoid missing any visual event in group boundaries.

The whole video stream,  $V$ , is divided into  $g$  overlapping groups to make the abstraction process faster and simpler. The video signal can be described by a combination of  $g$  video groups, defined as:

$$V = \bigcup_{i=1}^g G_i \quad (9)$$

where  $G_i$  represents the  $i^{\text{th}}$  group of the video sequence. Now considering  $Y$  as the feature matrix of the whole video sequence including frames 1 to  $N$ , it is defined as:

$$Y = [fv_1 \quad fv_2 \quad \dots \quad fv_j \quad \dots \quad fv_N] \quad (10)$$

Applying the grouping stage, the feature matrix of each group will be as follows:

$$Y_i = [fv_{i \times N_g - i \times N_{ov} + 1} \quad fv_{i \times N_g - i \times N_{ov} + 2} \quad \dots \quad fv_{(i+1) \times N_g - i \times N_{ov}}] \quad (11)$$

where  $Y_i$  is the matrix of parameters of the  $i^{\text{th}}$  group of frames. Considering the length of each group,  $N_g$ , and the number of overlapped frames between each group,  $N_{ov}$ , the  $i^{\text{th}}$  group contains frames  $i \times N_g - i \times N_{ov} + 1$  to  $(i + 1) \times N_g - i \times N_{ov}$ .

### 4.3. Video Temporal Decomposition

We then apply the TD method to each group  $G_i$  to detect event centroids of the corresponding parameter matrix  $Y_i$  of the  $i^{\text{th}}$  group using (1). Suppose that  $n_i$  is the number of the events and  $\text{ind}_{i,1}, \text{ind}_{i,2}, \dots, \text{ind}_{i,n_i}$  are the indices of the  $i^{\text{th}}$  group's event centroids selected by the TD method, and  $\text{fv}_{\text{ind}_{i,1}}, \text{fv}_{\text{ind}_{i,2}}, \dots, \text{fv}_{\text{ind}_{i,n_i}}$  are their corresponding feature vectors. These event centroids are employed as candidate frames for the keyframe selection process. Since the number of events extracted through the temporal decomposition is much smaller than the number of frames in the groups – usually less than 15% of the number of frames - a much smaller number of frames will be utilized in the keyframe extraction process.

The event centroids are taken as candidate keyframes but it is possible that these event centroids could be incorrect keyframes, since:

- 1) Perhaps there are the same visual events in the video group which occur in different parts of the group and they are selected as different events.
- 2) To avoid subdivision of long visual events, we consider these events as candidates of the keyframes, since some of the events may be too long, and similar frames may be selected as event centroids. We keep the length of the sub-blocks – temporal resolution - around the video temporal sampling rate to cover all visual events lasting around a second and keeping the computational complexity low. So, there may be redundancy in the detected events and all the event centroids may not be independent.

Thus to keep the algorithm simple and fast, we consider these event centroids as keyframe candidates and eliminate their redundancy in the proceeding stage described in 4.4.

### 4.4. Video Summaries

To select keyframes from among candidate frames of the  $i^{\text{th}}$  group, the mean distance between each candidate frame of group  $i$  and the other candidate frames in that group is calculated as:

$$D_i(j) = \frac{1}{n_i-1} \sum_{\substack{k=1 \\ k \neq j}}^{n_i} D(\text{fv}_{\text{ind}_i,j}, \text{fv}_{\text{ind}_i,k}) \quad (12)$$

where  $j = 1, \dots, n_i$  and  $D(\cdot)$  is the KLD between two feature vectors:  $\text{fv}_{\text{ind}_i,j}$  and  $\text{fv}_{\text{ind}_i,k}$ . Thus  $\frac{n_i}{2} \times (n_i - 1)$  calculations of  $D(\cdot)$  are required. Considering the fact that the number of event centroids is 15% of the number of frames in each group, where  $n_i = 0.15N$  and  $N$  is the number of frames, there will be approximately  $\frac{0.15N}{2} (0.15N - 1)$  calculations of  $D(\cdot)$ , which will be about  $0.0225N^2$  compared to the  $N^2$  calculations when all frames are involve in the keyframe selection step. In other words, the computational load will be reduced approximately  $\frac{1}{0.0225} = 44$  times compared with the methods which consider all frames.

The keyframes are the frames with the distance larger than a threshold  $T = 2m_w$ , that is predefined experimentally.  $m_w$  is the local mean of  $D_i$  on a  $w$  size window and is chosen equal to 3 here. So, the keyframes of each group are chosen as frames having the maximum distance from other event centroids in their groups.

Finally, the first and the last extracted keyframes of each group are compared to the keyframes of their adjacent groups and, if any two neighboring keyframes are similar, one of them is removed. The threshold used here is chosen as  $T = 0.5 \times (m_{g_i} + m_{g_{i+1}})$ , where  $m_{g_i}$  is the mean distance between the selected keyframes of group  $i$ .

The extracted keyframes are representatives of the video signal, which can improve the performance of several video processing applications, such as scene detection, analysis, editing and retrieval. They can also be utilized to manage huge databases and make the search process much faster and easier.

#### 4.5. Computational Complexity

To discuss about the complexity of the TD method, we first mention that the computational complexity of the SVD algorithm for a  $p$  by  $n$  matrix is  $O(pn^2)$  [25]. Thus, we can assume that a positive non-decreasing linear function  $f(\cdot)$  can be found so

that  $f$  is big-O of  $n^2$ . Accordingly, the complexity can be written as  $f(pn^2)$  and, due to its linearity, it can be rewritten as  $p \cdot f(n^2)$ . Applying the TD algorithm to the  $p \times n$  video matrix, the matrix is divided into  $p \times l_{sb}$  sub-blocks, thus the complexity of each sub-block is  $p \cdot f(l_{sb}^2)$ , and the total complexity of the TD-based analysis of the video group to extract event centroids is equal to  $n_{ev} \cdot p \cdot f(l_{sb}^2)$ . These event centroids are employed as keyframe candidates to extract the final abstraction. Consequently, the remaining part of the calculations in our method will be for the selection of keyframes from among  $n_{ev}$  candidate frames, as compared to  $n$  frames used for the keyframe selection by traditional methods.

We examine and compare two methods for keyframe selection to discuss the complexity of the idea. In the first method, the difference between feature vectors of adjacent frames are employed to select keyframes, thus the order of calculations can be represented by a positive non-decreasing linear function  $g(\cdot)$  -  $g$  is big-O of  $n$ . This leads the computational complexities to be  $g(n)$  and  $g(n_{ev})$  for the traditional and the proposed methods, respectively. The overall complexity of the proposed method can be calculated as  $n_{ev} \cdot p \cdot f(l_{sb}^2) + g(n_{ev})$ , where the experimental results show that  $n_{ev} < 0.15n$ , thus,  $g(n_{ev}) < 0.15g(n)$ , according to the linearity of  $g(\cdot)$ . The first part of the algorithm still needs more attention:  $n_{ev} \cdot p \cdot f(l_{sb}^2) < 0.15np_{max}f(n)$ , assuming the condition  $l_{sb}^2 < n$ .  $f(n)$  and  $g(n)$  are of the same order. They are also linear, positive, and non-decreasing functions. Therefore, a positive number  $a$  can be found so that  $f(n) \leq ag(n)$ . The overall complexity of our system can then be written as:

$$(0.15 + a (n_{ev})_{max} p_{max})g(n) \approx \left(0.15 + a \left(\frac{n}{l_{sb}}\right)_{max} p_{max}\right)g(n) \quad (13)$$

We want this statement to be smaller than the complexity of traditional methods,  $g(n)$ . This is achieved if the following reachable and simple conditions are satisfied:

$$l_{sb}^2 < n, \quad \frac{n_{max} \cdot p_{max}}{l_{sb_{min}}} < 0.85a \quad (14)$$

The second approach to keyframe extraction investigated here is to compare all frames within a certain segment of the signal and locate frames with bigger dissimilarities to other frames in the segment as the keyframes. In this case, the order of calculations can be represented by a positive non-decreasing linear function  $g(\cdot)$  -  $g$  is big-O of  $n^2$  - where the computational complexities will be  $g(n^2)$  and  $g(n_{ev}^2)$  for the traditional and the proposed methods, respectively. As mentioned above,  $n_{ev} < 0.15n$ , thus,  $g(n_{ev}^2) < 0.0225g(n^2)$  and the total complexity of our method will approximately be equal to  $10^{-2}f(n^2) + 10^{-2}g(n^2) \leq 10^{-2}(1 + a)g(n^2)$ , assuming that  $l_{sb} \approx \frac{n}{n_{sb}}$  and  $n_{ev} \cdot p \approx 1$ , where  $n_{sb}$  is the number of sub-blocks in the group and is always greater than 10. Again, due to the fact that the linear, positive and non-decreasing functions  $f(n^2)$  and  $g(n^2)$  are of the same order, a positive number  $a$  can be found so that  $f(n^2) \leq ag(n^2)$ . Therefore, the overall complexity of the proposed scheme is smaller than that of the traditional keyframe selection methods, if  $a < 99$ ; and if this condition is not valid, the condition  $l_{sb}^2 < n$  can be applied to eliminate the first sentence of the inequality and make the equation valid. However  $a$  is much smaller than this threshold since both functions are big-Os of  $n^2$ . So the overall complexity of the proposed approach will at less than 10% of the traditional method. This is while the computational load of shot and shot cluster boundaries selection stage in traditional methods have not been considered here: considering those will lead to an even lesser complexity of our method compared with traditional methods.

## 5. EXPERIMENTAL RESULTS

The proposed method has been implemented and evaluated in various keyframe extraction experiments. Here, we give details of the experiments and the major results attained.

## 5.1. Test Set

We have employed a large number of natural video sequences of more than 10 hours video with sampling rates of 15, 24 and 29 frames per seconds. Most of the videos are selected from Hollywood2 Human Actions and Scenes dataset [14], the TRECVID dataset [17], the ‘Simon Fraser University (SFU) Video Library and Tools dataset’ [15], and ‘The Open Video Project database’ [16]. The size of each frame is between QCIF (176x144) and CIF (352x288). The video tests are captured from different locations – indoor and outdoor- and of different characteristics. They display car racing, news broadcasting, dogs running, plane flying, glass being broken, etc. Also, they contain camera zooming, panning, translations and scene dissolves. Most of the test videos are very dynamic in both temporal and spatial domains, though there are also few static samples. To select the preliminary values of some parameters, such as  $l_{sb}$ ,  $N_g$ ,  $N_{ov}$  and the thresholds, we have used the ‘Hollywood2 Human Actions and Scenes dataset’. This dataset contains 1152 video sequences of 1,025,278 video frames and 8199 shot transitions collected from more than 69 movies. The proposed method is applied to these videos and the results are evaluated manually and the best values are selected based on the abstraction results from these videos.

**Table 1.** Details of employed video data set of TRECVID 2006.

Video	Video name	# of frames	# of shots	Shot ratio	Mean frames/shot
1	20051101_142800_LBC_NAHAR_ARB.mpg	112087	243	0.22%	461
2	20051114_091300_NTDTV_FOCUSINT_CHN.mpg	31138	198	0.64%	157
3	20051115_192800_NTDTV_ECONFRNT_CHN.mpg	31169	168	0.54%	186
4	20051129_102900_HURRA_NEWS_ARB.mpg	22659	148	0.65%	153
5	20051205_185800_PHOENIX_GOODMORNCN_CHN.mpg	58142	360	0.62%	162
6	20051208_125800_CNN_LIVEFROM_ENG.mpg	58142	367	0.63%	158
7	20051208_145800_CCTV_DAILY_CHN.mpg	58142	441	0.76%	132
8	20051208_182800_NBC_NIGHTLYNEWS_ENG.mpg	58142	638	1.1%	91
9	20051209_125800_CNN_LIVEFROM_ENG.mpg	12864	94	0.73%	137
10	20051213_185800_PHOENIX_GOODMORNCN_CHN.mpg	58142	271	0.47%	215

11	20051227_105800_MSNBC_NEWSLIVE_ENG.mpg	58142	520	0.89%	112
12	20051231_182800_NBC_NIGHTLYNEWS_ENG.mpg	28502	266	0.93%	107
13	20051227_125800_CNN_LIVEFROM_ENG.mpg	9772	71	0.73%	138
Total		597043	3785	0.68%	158

To assess the efficiency of the proposed video abstraction algorithm, we specifically used the TRECVID 2006 shot boundary detection video database. This library contains 13 long videos of 597,043 video frames, where there are 3785 shot transitions - including 48.7% cut transitions and 51.3% gradual transitions - and the temporal sampling rate is 29 frames per second. More information about this video set is shown in table 1, where ‘shot ratio’ shows the percentage of shots per the video frames and ‘mean frame/shot’ is the average number of frames per shot. As mentioned in the table, most of the video samples are dynamic and the ‘mean frames/shot’ is about 158 frames, which shows there are many short shots and some long shots in the video samples, especially in video sample 1.

## 5.2. Video Abstraction Results



(a) Temporally downsampled video sequence.



(b) Extraction results based on the proposed method (GGD spatial parameters + TD).



(c) Extraction results based on color



(d) Extraction results based on color

features (HSV spatial parameters + TD).

features (HSV spatial parameters +  
Euclidean distance).

**Figure 2.** Keyframe extraction results using the proposed method.

As per the process explained in section 4, we have extracted the GGD parameters from the 2D wavelet transformed subbands of each frame, and constructed the parameter matrix for the videos. Next, grouping is applied to the long videos and the event centroids of each video group are located by the TD. Finally, the keyframes are selected from among the event centroids of each group, where similar neighboring keyframes of adjacent groups are removed. It should be mentioned that the length of each group is  $N_g$  and there is an overlap of  $N_{ov}$  - frame between adjacent groups.



(a) Temporally downsampled video sequence.



(b) Extraction results based on the proposed method (GGD spatial parameters + TD).



(c) Extraction results based on color features (HSV spatial parameters + TD).



(d) Extraction results based on color features (HSV spatial parameters + Euclidean distance).

**Fig. 3.** Comparison of keyframe methods.

To evaluate our method, we use the color histogram features based on histograms in the Hue Saturation and Value (HSV) color space. HSV has been used in many video processing applications as a representation of visual contents of frames [4,13]. In such methods, normalized HSV histograms are calculated and eight, four, and four features are extracted from Hue, Saturation, and Value coordinates, respectively. To calculate the distance between HSV feature vectors, Euclidean distance measure is employed. The keyframe extraction examples are shown in figures 2 and 3. The frames are temporally downsampled. As shown in the figures, the keyframes extracted using our proposed method (part *b* in each figure) represent the video sequences more clearly than the other methods and detect dissimilarities properly.

To show the efficiency of our employed spatial features, the HSV color features are used and keyframes are selected through our method. The results of this approach are depicted for two examples in figures 2.c and 3.c. Furthermore, to compare the proposed method to major existing techniques, the HSV color feature vectors are employed and shots, shot clusters, and keyframes are selected based on Euclidean distance measure. The resulting keyframes are shown in part *d* of figures 2 and 3. As mentioned earlier, color features have been used in many applications [4, 13]. Our method outperforms the HSV method, since it can identify details that are contextually more important than colors. For the example shown in figure 3, some of the frames have no representative keyframe in parts *c* or *d*; for instance the frames in the third row of figure 3.a. But with our method a more accurate keyframe selection is achieved, because the wavelet space comes with a better approximation to the HVS [8].

We have also conducted a user study to evaluate our method, in addition to the comparison made to the HSV methods, as there is no standard evaluation method for keyframe extraction [18]. We asked 9 independent subjects having some knowledge about video processing methods to rate results of the keyframes extraction method as ‘Good’, ‘Acceptable’ or ‘Bad’ based on the method used in [19]. Each time, a video shot is extracted manually and is displayed to the subject in the top part of the screen and the extracted keyframes are shown in the down part of the window. Then the subject is asked to judge the quality of keyframe extraction method. Next, the subsequent video shot and its results are displayed to the user and the procedure is repeated till the end of the video clip.

**Table 2.** Subjective evaluation results on the TRECVID database: #Gr: number of groups, #Sh: number of shots, #Ev: number of detected events, #Kf: number of selected keyframes.

System parameters : Wavelet filter: ‘Daubechies4’, Transform levels: 4, Group length ( $N_g$ ): 250, Overlap between groups ( $N_{ov}$ ): 25, sub-block length ( $l_{sb}$ ): 25.

	#Gr	#Sh	#Ev	#Kfs	Good	Acceptable	Bad	Kf%	Kf/shot
1	498	243	16518	323	215(88.48%)	15(6.17%)	13(5.35%)	0.29%	1.329
2	138	198	4870	332	162(81.82%)	17(8.59%)	19(11.73%)	1.07%	1.677
3	138	168	4808	193	141(83.93%)	12(7.14%)	15(10.64%)	0.62%	1.149
4	100	148	3436	231	123(83.11%)	17(11.49%)	8(6.50%)	1.02%	1.561
5	258	360	8730	472	284(78.89%);	43(11.94%)	33(11.62%)	0.81%	1.311
6	258	367	8367	545	327(89.10%)	25(6.81%)	15(4.59%)	0.94%	1.485
7	258	441	8655	767	363(82.31%)	42(9.52%)	36(9.92%)	1.32%	1.739
8	258	638	8618	878	544(85.27%)	47(7.37%)	47(8.64%)	1.51%	1.376
9	57	94	1816	192	83(88.30%)	8(8.51%)	3(3.61%)	1.49%	2.043
10	258	271	8744	396	216(79.70%)	33(12.18%)	22(10.19%)	0.68%	1.461
11	258	520	8898	617	463(89.04%)	15(2.88%)	42(9.07%)	1.06%	1.187

12	126	266	4287	447	219(82.33%)	26(9.77%)	21(9.59%)	1.57%	1.68
13	43	71	142	115	62(87.32%)	4(5.63%)	5(8.06%)	1.18%	1.62

It should be noted that this evaluation method based on each video shot and the correspondence keyframes does not change our algorithm process. We have extracted the keyframes based on the group based method and no shot boundary detection has been applied. This is the subjective test procedure which displays the video clips and results per shot. Table 2 contains the subjective tests results on the TRECVID database, where ‘Kf%’ represents the percentage of keyframes over all video frames, and ‘Kf/shot’ is the mean number of keyframes per shot. Results show that ‘Kf%’ is high in the videos which have short shots or are more dynamic in the scenes (video samples 7 and 8). Also ‘kf/sh’ is low in the videos which have low activity shots (video sample 1). This confirms the ability of our method to capture the temporal evolution of visual contents of the video signal. Moreover, the rating results show the percentage of ‘Good’ rating is 85% on average which is a sign of the effectiveness of our model to choose and locate the keyframes accurately.

We have also compared our method to three other methods in another subjective test. In this subjective test, in addition to the described HSV color features, L\*a\*b\* (CIELab) color features are also employed. L\*a\*b\* color space is designed based on a large number of subjective experimental tests and covers all visible colors, with the known fact that the Euclidean distance is a suitable measure to show the accurate perceptual difference between colors of two images in this space [26]. Like the HSV features, the normalized color histograms of each color space is calculated and eight, four and four features are extracted from L\* (lightness), a\* (position between red/green) and b\* (position between blue/yellow) coordinates respectively. Also the Euclidean distance is used to determine the distance between two feature vectors. The video sample 9 from the TRECVID database was selected and the rating procedure was repeated as describe earlier. Table 3 contains the evaluation results which show the efficiency of our method.

It confirms that the GGD features extracted from the wavelet transform of the frames contain more important information for HVS, compared to the color features.

Methods are defined in the table as follows:

Method 1: GGD spatial parameters + TD.

Method 2: HSV spatial parameters + TD.

Method 3: HSV spatial parameters + Euclidean distance.

Method 4: L\*a\*b\* spatial parameters + Euclidean distance.

**Table 3.** Subjective evaluation results on the video sample 9. #Frames:12864,  
#Groups:57, #Shots:94, #Fr/sh:132.

	#Events	#Keyframes	Good	Acceptable	Bad	Kf%	Kf/sh
Method1	1816	192	83(88.30%)	8(8.51%)	3(3.61%)	1.49%	2.043
Method2	1720	174	75(79.79%)	11(11.70%)	8(8.51%)	1.35%	1.851
Method3	-	153	70(74.47%)	14(14.89%)	10(10.64%)	1.19%	1.628
Method4	-	157	72(76.60%)	11(11.70%)	11(11.70%)	1.22%	1.670

As mentioned in section 4, one of the advantages of employing our simple and fast scheme is that no shot and cluster boundary selection is required, and the algorithm finds the event of each group separately and ignores similar keyframes extracted on the boundaries of adjacent groups. This grouping structure provides the capability to use our method in real-time applications.

Since we expect to have at least one keyframe selected from each video shot, we have performed several tests on the TRECVID dataset, varying the system parameters – different wavelet filters and levels, different group, overlap between adjacent groups and sub-block lengths. The total numbers of frames and shots in the database are 597043 and 3785, respectively. The results are shown in tables 4 to 8, where ‘P’ is the measure showing the percentage of shots which have at least one corresponding keyframe and has been selected as an assessment to confirm the precision of the system in capturing temporal evolution of the video contents, and ‘Event%’ is the percentage of event

centroids over all frames. It is a measure defined to show the average ratio of candidate frames (event centroids) to all frames. This number helps us to evaluate and show the benefit of complexity reduction of our method.

In table 4, the effect of group size is studied. Keyframe per shot, Keyframe percentage, and ‘P’ increase as the group size decreases, while the length of groups and the computational load increase.

**Table 4.** Evaluation results for different values of Group length ( $N_g$ ). Wavelet filter: ‘Daubechies’, Transform levels: 4, Overlap between groups ( $N_{ov}$ ): 25, sub-block length ( $l_{sb}$ ): 25.

$N_g$	#Groups	#Events	#Keyframes	Kf/sh	Kf%	Event%	P
150	4773	96284	6549	1.7303	1.10	0.1613	95.25%
200	3406	91378	5765	1.5231	0.97	0.1531	94.15%
250	2648	89175	5508	1.4552	0.92	0.1494	93.70%
300	2164	87009	5113	1.3509	0.86	0.1457	92.56%

The effect of the number of overlapping frames  $N_{ov}$  on the abstraction results is given in table 5. As shown, keyframe per shot, keyframe percentage, and ‘P’ increase slightly by increasing the number of overlapping frames. Thus, the results are not much affected by changing  $N_{ov}$  in the given range and a number near 15 could be reasonable.

**Table 5.** Evaluation results for different numbers of overlapped frames ( $N_{ov}$ ). Wavelet filter: ‘Daubechies’, Transform levels: 4, Group length ( $N_g$ ): 250, sub-block length ( $l_{sb}$ ): 25.

$N_{ov}$	#Groups	#Events	#Keyframes	Kf/sh	Kf%	Event%	P
15	2534	85364	5166	1.3649	0.87	0.1430	92.91%
20	2587	87134	5288	1.3971	0.89	0.1459	93.27%
25	2648	89175	5508	1.4552	0.92	0.1494	93.70%

Table 6 contains the evaluation results for different sub-block sizes, where precision of the method, keyframe percentage, event percentage, and keyframe per shot are inversely related to  $l_{sb}$ . Decreasing the temporal resolution, more short visual events are considered and the system better detects short shots.

**Table 6.** Evaluation results for different sub-block sizes ( $L_{sb}$ ). Wavelet filter: ‘Daubechies’, Transform levels: 4, Group length ( $N_g$ ): 250, Overlap between groups ( $N_{ov}$ ): 15, Number of groups: 2534.

$l_{sb}$	#Events	#Keyframes	Kf/sh	Kf%	Event%	P
15	134355	5811	1.5353	0.97	0.2250	93.23%
25	85364	5166	1.3649	0.87	0.1430	92.91%
35	58813	4556	1.2037	0.76	0.0985	91.67%
45	44150	4015	1.0608	0.67	0.0739	90.94%

**Table 7.** Evaluation results for different number of wavelet decomposition levels.

Wavelet filter: ‘Daubechies’, Group length ( $N_g$ ): 250, Overlap between groups ( $N_{ov}$ ): 25, sub-block length ( $l_{sb}$ ): 25, Number of groups: 2648.

Levels	#Events	#Keyframes	Kf/sh	Kf%	Event%	M1
3	89868	4724	1.2481	0.79	0.1505	91.48%
4	89175	5508	1.4494	0.92	0.1494	93.70%
5	90169	6696	1.7691	1.12	0.1510	96.35%

**Table 8.** Evaluation results for different wavelet filters. Decomposition levels: 4, group length ( $N_g$ ): 250, overlap between groups ( $N_{ov}$ ): 25, sub-block length ( $l_{sb}$ ): 25, number of groups: 2648.

Levels	#Events	#Keyframes	Kf/sh	Kf%	Event%	P
Haar	89436	4791	1.2658	0.80	0.1498	92.52%

Daubechies4	89175	5508	1.4494	0.92	0.1459	93.70%
Symlet4	90862	6547	1.7297	1.10	0.1522	95.95%

We have also studied the effect of the spatial features on the efficiency of the scheme in tables 7 and 8. Increasing the transform levels, more detailed information is extracted from frames and the results improve. Furthermore, the results are relative to the wavelet filter and the best results are achieved applying ‘Symlet’ filter which is a symmetric filter.

The above tests confirm the high efficiency of the proposed method in video abstraction. This high efficiency can be attributed to employing two important factors along with the benefit of utilizing temporal decomposition and grouping: 1) Choosing appropriate wavelet features (GGD features) which match closely with the Human Visual System [9, 10] characteristics and 2) Using a suitable distance measure (KLD measure).

Considering the fact that no shot boundary extraction step is needed in our group based approach, this scheme could be the choice for a relatively fast, efficient, and accurate scene analysis in various video processing applications. The experimental results are summarized in table 9, where the effect of various parameters are shown in the results and the best choices for these parameters are selected.

**Table 9.** Effect of the different parameters on the abstraction results.

	Kf/sh	Kf%	P	Computational load	Best match
$N_g \uparrow$	↓	↓	↓	↑	200-250
$N_{ov} \uparrow$	↑	↑	↑	↑	20-25
$l_{sb} \uparrow$	↓	↓	↓	↓	15-20
Transform level↑	↑	↑	↑	↑	4

## 6. CONCLUSION

We have presented a group based spatio-temporal method for natural video analysis that uses Temporal Decomposition (TD) to locate visual events. The video signal is taken to be a sequence of overlapped visual events. The TD describes the temporal evolution of this signal through a linear approximation to the visual events generating the video sequence. The GGD parameters, extracted from marginal statistics 2D wavelet transformed subbands of each frame, are used as spatial parameters in the TD to extract the events. To reduce the processing power and save time, first the video stream is divided into successive overlapped groups of frames. Unlike conventional keyframe selection methods, no processing is required to determine shot or shot cluster boundaries. Furthermore, the grouping stage is simple and reduces the computational complexity of the TD and gives the opportunity to use the method in real-time applications. We have also shown, through scene analysis and extensive subjective tests, that the proposed approach outperforms the conventional keyframe extraction methods significantly.

## 7. REFERENCES

- [1] Zhuang, Y., Rui, Y., Huang, T.S., Mehrotra, S.: Adaptive key frame extraction using unsupervised clustering. *IEEE Int'l Conf. on Image Processing*, pp. 283-287 (1998). doi: 10.1109/ICIP.1998.723655
- [2] Nagasaka, A., Tanaka, Y.: Automatic video indexing and full-video search for object appearances. *Visual Database Systems II*, Elsevier, 15(2) pp. 113-127 (1992)
- [3] Bulut, E., Capin, T.: Key frame extraction from motion capture data by curve saliency. *Computer Animation and Social Agents, CASA*, (2007)
- [4] Chen, W., Zhang, Y.J.: Parametric model for video content analysis. *Elsevier B.V. Pattern Recognition Letters*. 29:181–191 (2008). doi: 10.1016/j.patrec.2007.09.020
- [5] Manor, L.Z., Irani, M.: Event-Based Video Analysis. *IEEE Conf. on Computer Vision and Pattern Rec.* 2:123-130 (2001). doi: 10.1109/CVPR.2001.990935
- [6] Polana, R., Nelson, R.: Detecting activities. *IEEE Conf. on Computer Vision and Pattern Recognition* pp. 2-5 (1993). doi: 10.1109/CVPR.1993.341009
- [7] Atal, B.S.: Efficient coding of LPC parameters by temporal decomposition. *IEEE Int'l Conference on Acoustics, Speech, and Signal Processing, ICASSP*, pp. 81-84 (1983). doi: 10.1109/ICASSP.1983.1172248
- [8] Ghaemmaghami, S.: Audio segmentation and classification based on a Selective analysis scheme. *10th Int'l Multimedia Modelling Conference, MMM*, pp. 42-47 (2004). doi: 10.1109/MULMM.2004.1264965
- [9] Mallat, S.: A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. Patt. Recog. and Mach. Intell.* 11(7):674– 693 (1989). doi: 10.1109/34.192463

- [10] Oh, T.H., Besar, R.: JPEG2000 and JPEG: image quality measures of compressed medical images. 4<sup>th</sup> National conf on Telecom. Tech., NCTT Proceedings, pp. 31 – 35 (2003). doi: 10.1109/NCTT.2003.1188296
- [11] Simoncelli, E.P., Duccigrossi, R.W.: Embedded wavelet image compression based on a joint property model. IEEE Int'l Conf. Image Processing, VoU, 1:640-643 (1997). doi: 10.1109/ICIP.1997.647994
- [12] Do, M.N.: Directional multiresolution image representations. PhD thesis. Swiss Federal Institute of Technology (2001)
- [13] Pickering, M.J., Ryger, S.: Evaluation of key frame-based retrieval techniques for video. Elsevier Computer Vision and Image Understanding, CVIU, 92(2-3):217-235 (2003). doi: 10.1016/j.cviu.2003.06.002
- [14] <http://www.irisa.fr/vista/Equipe/People/Laptev/download.html> (2011). Accessed 15 april 2011
- [15] [http://nsl.cs.sfu.ca/wiki/index.php/Video\\_Library\\_and\\_Tools](http://nsl.cs.sfu.ca/wiki/index.php/Video_Library_and_Tools) (2011). Accessed 15 april 2011
- [16] <http://www.open-video.org> (2011). Accessed 15 april 2011
- [17] <http://www-nlpir.nist.gov/projects/trecvid> (2011). National Institute of Standards and Technology (NIST). Accessed 15 april 2011
- [18] Truong, B.T., Venkatesh, S.: Video Abstraction : A Systematic Review and Classification. ACM Trans. Multimedia Comput. Commun. Appl. 3(1) (2007). 10.1145/1198302.1198305
- [19] Liu, T., Zhang, H.J., Qi, F.: A novel video Key-frame extraction algorithm based on perceived motion energy model. IEEE Trans. On Circ. Sys. And Video Technology 13(10):1006-1013 (2003). doi: 10.1109/TCSVT.2003.816521
- [20] Truong, B.T., Venkatesh, S.: Video abstraction: A systematic review and classification. ACM Trans. Multimedia Comput. Commun. Appl. 3:1-37 (2007). doi: 10.1145/1198302.1198305
- [21] Li, Y., Lee, S.H., Yeh, S.H., Kuo, C.-C.J.: Techniques for Movie Content Analysis and Skimming. IEEE Signal Processing Magazine 23:79-89 (2006). doi: 10.1109/MSP.2006.1621451
- [22] Janvier, B., Bruno, E., Pun, T., Maillet, S.M.: Information-Theoretic Temporal Segmentation of Video and Applications: Multiscale Keyframes Selection and Shot Boundaries Detection. Multimedia tools and application 3(3):273-288 (2006). doi: 10.1007/s11042-006-0026-2
- [23] Cooper, M.L., Foote, J.: Discriminative techniques for keyframe selection. Int'l Conference on Multimedia and Expo, ICME, pp. 502-505 (2005). doi: 10.1109/ICME.2005.1521470
- [24] Shao, L., Ji, L.: 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). doi: 10.1109/CRV.2009.36
- [25] Manjunath, B.S., Chandrasekaran, S., Wang, Y.F.: An eigenspace update algorithm for image analysis. Int. Symp. on Computer Vision, pp.551-556, (1995). doi: 10.1109/ISCV.1995.477059
- [26] Cover, T.M., Thomas, J.A.: Elements of Information Theory. New York, Wiley (1991)
- [27] Jain, A.K.: Fundamentals of Digital image Processing. Prentice Hall, ISBN-13: 9780133361650 (1989)