Journal of Computer Science 2 (4): 326-332, 2006
ISSN 1549-3636

An Effective Transmission and Browsing Methodology for Streaming Video

[1]K.K. Thyagharajan and [2]V. Ramachandran
[1]SSN College of Engineering, Kalavakkam- 603 110, India
[2]College of Engineering, Guindy, Chennai, 600 025, India

Abstract: Browsing streaming video quickly through Internet with in the available bandwidth is a challenging one and it is also time consuming. The user has to browse, view and listen to the complete video to identify relevant contents. Most of the video summarization techniques presented in research papers are suitable only for stand-alone environments. They are not addressing or analyzing the problems faced by streaming video and do not provide an evaluation system to evaluate streaming parameters. The objective of video segmentation and summarization is to maximize the information rate from a streaming server to client in media access activities. This study discusses the segmentation techniques for creating video summary and proposes a hierarchical scheme, which decomposes a video sequence into different content-resolution levels to enhance the transmission and user interaction. Mathematical models have been derived to represent the video structure. Hence, streaming parameters such as bandwidth, buffer requirements and initial delay are evaluated for each segment at different levels to provide jitter-free playback. The algorithm developed in this study can be attached with a video encoder to evaluate streaming parameters and the result can be provided to a streaming server to create an effective transmission schedule.

Key words: Segmentation, summarization, mathematical models, video analysis

INTRODUCTION

Video is an information-intensive media with much redundancy. Therefore browsing streaming video quickly through the Internet with in the available bandwidth is a challenging one. In comparison with browsing text in which quick glance is sufficient to filter information, browsing a video is much more time consuming. This is because the user has to browse, view and listen to the complete video to identify relevant contents. The VCR like control features such as fast-forward and fast-rewind do not help much because those controls search the video in a sequential manner. It occupies the complete bandwidth allotted and also the audio is not distinguishable during these operations. Since video does not have a hierarchical structure in terms of content, it is very difficult to search for a specific content in a video. Therefore for transmitting the required video efficiently, the client should be allowed to search or browse in a non-sequential manner from a video arranged in a hierarchical order. Researches show that it is very difficult to search for a specific subtopic in an unstructured video. To create a structure, the video must be indexed and segmented.

Low level video indexing methods such as scene or shot detection use primitive or low-level features such as object motion, color, texture, shape and spatial location. High level content-based indexing method uses video annotation or meta-data. This method uses the semantic features of video at various degrees and annotates them using a video annotating tool. Systems during nineties did not use semantic information in the video[1]. By 1993, the use of Image Processing techniques leads to content-based solution. The low-level techniques are automatic techniques but they do not provide semantic information.

The other method for indexing the video is to describe the video segments with text using an annotation tool or representative key frames and use them for content-based retrieval. Annotations play an important role in describing raw data from various points of view and enhancing the query process. Annotations are subjective comments, explanations or external remarks that attached to a selected part of a document without modifying the document[2]. From annotations, index points and meta-data can be created for content-based retrieval. Using the index points, video is summarized. Video summarization represents an entire video clip into a more compact movie. A pictorial video summary enables viewers to grasp main contents of a video at a glance.

Content-based video summarization allows users to browse and retrieve the desired video segments in a non-sequential fashion[3]. A hierarchical summary of video conveys visual content at various levels of detail. Jia-Yu Pan *et.al.*,[4] proposed a technique in which a video is broken into shots and each shot is associated with a still frame (key frame) and transcript words. Transcript words are used for content-based summarization. Dulce Ponceleon[5] presents a summarized representation of video content using key

Corresponding Author: K.K. Thyagharajan, SSN College of Engineering, Kalavakkam- 600 011, India

frames and calls it a storyboard. A storyboard represents several minutes of video on a single HTML page. The key frame used for indexing the video in a video summary exposes the content of video and the user can view it for browsing.

A meta-data is a collection of descriptors that tells something about the video content and it can be used as index terms for video browsing and deliver it in a manageable format. Semantic meta-data schemes based on XML descriptions are used to interactively annotate videos. Piera Palma[6] creates index based on visual and textual content of video by using virtual image and meta-data. DamgSong proposes a natural language approach to content-based video indexing and retrieval[7]. Nikolaos[8] presents an interactive framework for navigating video sequences over IP-based networks using an optimal content-based video decomposition scheme.

This study reviews the video segmentation techniques in section II and finds that there is no means to evaluate them with streaming video. This study proposes a hierarchical segmentation scheme for video, which combines both low-level and high-level segmentation methods and provides a means to the user to search for the desired video segment from a video summary. This hierarchical video summary allows the user to browse the video at different content-resolution levels. The user can select a specific video segment and zoom it at higher content resolution. The main objective of the video summarization is to maximize the information transfer from streaming server to the client's browser using minimum network resources. Mathematical models derived in this study to represent each segment in the hierarchically structured video help to evaluate the streaming parameters such as buffer and minimum bandwidth required at different levels in the segment hierarchy. These parameters can be used to do effective streaming, which reduces the bandwidth and buffer requirements while searching and playing the video content.

Video segmentation techniques: Video segmentation is the first step to analyze video content[5]. Video segmentation techniques can be classified into two broader categories such as low-level segmentation and high-level segmentation. The low-level segmentation is used to automatically organize the syntactic structure[2] such as scenes, shots and key frames. Automatic extraction algorithms work well for low-level features such as color histogram, shape, texture and motion[10]. But this requires extensive image processing. In the proposed video structure model, low-level segmentation and indexing techniques can be used to divide the video into scenes, shots and Group Of Pictures (GOPs). There is little firm evidence that current low-level indexing techniques are adequate for multimedia repository exploration. According to Chabane [11] low-level indexing in video databases, art galleries and museums is still an open problem.

Each element in the video structure is represented by a key frame or a video skim. Video skim is a shorter version of video[12,13]. The representational power of a key frame or a skim depends on how it is chosen from all frames of a sequence. For example, a blurring frame caused by fast camera motion or object motion is not a good candidate for reference frame. The frame with low motion intensity must be selected. Generally, the middle frame is chosen to avoid unexpected effects near the boundaries. The representative key frame should be more informative and should be distinctive from each other. Image quality should be as high as possible. An inter-coded frame requires a preceding key frame and may not be used as a representative frame and hence a key frame preceding the inter-coded frame is chosen as a representative frame.

A shot is a contiguous sequence of video frames recorded from a single camera operation. Shots are actual physical layers in video, whose boundaries are determined by editing points where camera switches. A scene is a series of shots logically connected but need not be contiguous. There is no universal definition and rigid structure exists for scenes. Automatic shot detection tools use many techniques to separate and index a shot. They generally recognize cuts, dissolves and fades to detect a shot. A cut is a clean transition between a current shot and the following shot[6]. The producer structures most of the shots to progress smoothly from the beginning to the end. Therefore, it is reasonable to extract a key frame based on the duration of the shot. Even though this approach is not perfect, it is well suited to typical feature film. Abrupt scene changes and static scene detection is based on a comparison of DCT coefficients or motion vectors of subsequent frames. The number of representative frames can be reduced by filtering out noisy, blurry, uncolored and repetitive frames. For example, in a dialogue scene both speakers will be shown several times but it can be represented using two frames. Video can also be segmented by keywords (text) that are automatically extracted from video images using OCR software or from sound track.

The high-level segmentation requires findings of events, objects and scenes with high-level interpretation according to domain knowledge. In[11], Chabane interprets high-level features as logical, derived and semantic features and they are subjective features. Persons who index the video annotate these high-level features manually. These subjective features concern abstract attributes and describe the meaning of objects or scenes. Content description such as object descriptions, event description other lexicon sets and own keywords are added in the form of meta-data. A lexicon is an MPEG-7 based definition of application dependent description components that has no standardized format. The structural and content attributes extracted in feature extraction, abstraction

processes or the attributes that are entered manually are often referred to as meta-data. The object recognition feature requires manual annotation or manual content description and content-based video segmentation still requires human assistance[14,15]. To describe shots in the content-based segmentation, they can be categorized as events, activities, emotions, actions and dialogues, indoor or outdoor. Looking for a specific event in a video has still remained a far-reaching goal. One approach is to present a summarized representation of the video content and let the users spot the event.

A hierarchical structure of video: The streaming video is not hierarchically structured; it has a sequential structure based on a linear time line. The interaction between viewer and video is based on the fast-forward and rewind functions. A video-on-demand (VoD) system provides a navigation menu, but this provides facility to select a specific video clip without providing the complete details of the video. Process of creating a presentation about the structure of video is called Video Abstraction[14], Story Board or Cataloging[5], or Summarization[4,16]. As pointed out by Daniel in[17] a video summary should concisely present the contents of the input video source. It should be shorter than the original, focus on the content and give the viewer an appropriate overview of the whole. Generally, a video summary offers the user the basic information at a glance on what is happening in the video and ability to decide whether to play it and what part of it is to play. Video segmentation is the first step to analyze the video content. According to Yu-Fei Ma[18], structure analysis in video is still an open issue.

This section proposes a segmentation scheme that decomposes a video sequence into different content-resolution levels as shown in Fig. 1. The number of representative frames determines the content-resolution. This depends on the complexity of the examined video sequences i.e. more complicated sequences demand larger number of representative shots or frames. The video model developed in this section helps to specify various parts of the video structure and establish relationship among them like a book. The hierarchical structure of video acts as a visual table of contents of a book. According to Nevenka[14], the visual table of contents built based on structure information and key frames, provide an ideal representation for video browsing. The segments, scenes, shots and Group Of Pictures (GOP) can be compared with chapters, paragraphs, sentences and words of a book respectively. This also acts as a compact video summary and enables the user to quickly browse through video sequence and locate segments of interest. This structure allows the users to preview video sequences at various resolutions and zoom in on the segments of their interest. The elements at the bottom of the structure have highest or fine content resolution i.e. it contains the maximum representative frames. The GOPs are represented as level 1 elements, shots are level 2 elements and so on. Low-level automatic segmentation techniques can be used to divide the video into scenes, shots and GOPs. The relationship between segments and other elements can be established either automatically using data mining concepts or manually using annotating tools. Annotations play an important role in describing video from various points of view and in enhancing the retrieval process. The objective of video summarization is to maximize the information rate from the streaming server to the user in media access activities.

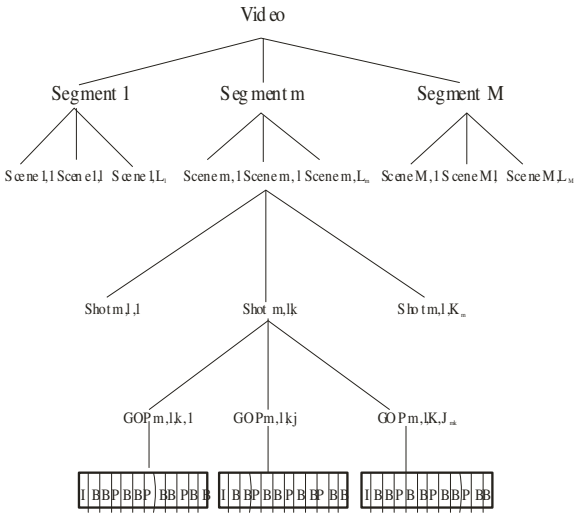

Fig. 1: Hierarchical structure of video at different content-resolutions

This hierarchical structure can be implemented using XML format as shown below. This is a high level representation.

```
<video>
    <segment>
        <scene>
            <shot>
              <GOP>
                  <frame>15532</frame>
              </GOP>
            </shot>
        </scene>
    </segment>
</video>
```

A hierarchical representation of a video is obtained by using every N^{th} frame with different values of N at various levels of hierarchy[16]. Figure 2 shows a Binary Tree structure for streaming video summarization. In this structure, N represents the total number of frames in the video. This structure is created automatically without human assistance. The representative frames (N/2, N/4, etc.,) represent the contents of the video to provide an estimate about the video contents to the user. When the number of levels increases, the user gets more of the video contents exposed. In this approach, a

single connection or session is established between the streaming server and the client and each representative frame uses a separate stream at lower bit-rate transmission. Each representative key frame has time index; when the user clicks on that frame, the video will play starting from the time pointed by the time index.

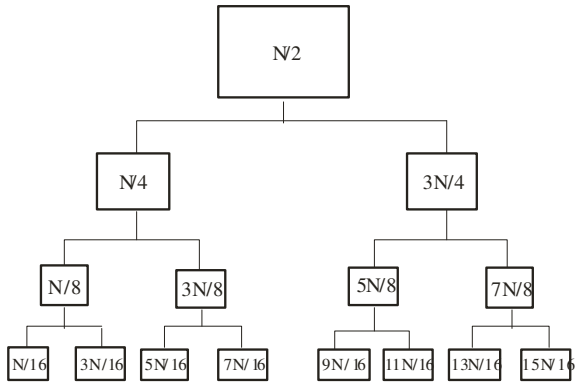

Fig. 2: Binary tree structure of streaming video summarization

The proposed hierarchical structure of video given in Fig. 1 is flexible because, any number of segments can be added in each level and hence the number of levels is minimized. Each level is also directly related to the structure of a book and can be used to create content-based summarization with minimum manual assistance.

Mathematical modeling: Let V be a video, which has *n* number of frames. If any frame f_i indicates the size of that frame in bytes, then the size of V is given as

$$V = \sum_{i=1}^{n} f_i \quad (1)$$

Let V be divided into M number of segments. If the total number of bytes in segment *m* is represented as E_m,

$$V = \sum_{m=1}^{M} E_m \quad (2)$$

The size of E_m is calculated by adding the size of all frames in that segment. If all the segments are having equal probability ^pM for being selected, then

$$^pM = 1/M \quad (3)$$

Hence, the average number of bytes transferred for viewing a segment is given by

$$E_{(av)} = \sum_{m=1}^{M} E_m * {}^pM$$

$$= \frac{1}{M} \sum_{m=1}^{M} E_m = \frac{V}{M} \quad (4)$$

If the total bytes in the video take T seconds to playback i.e. the length of the video is T seconds, then the average bandwidth Bv(av) required for transmitting the video is given by

$$B_{v\,(av)} = \frac{V}{T} \text{ bytes/sec} \quad (5)$$

If the transmission of the video starts d seconds before the playback starts i.e. if the initial delay is d seconds, then

$$B_{v\,(av)} = \frac{V}{T + d} \text{ bytes/sec} \quad (6)$$

$$B_{v\,(av)} = \frac{M * E_{(av)}}{T + d} \text{ bytes/sec} \quad (7)$$

Let

$$d = \zeta T : \zeta \geq 0 \quad (8)$$

where ζ is a constant

$$B_{v\,(av)} = \frac{M * E_{(av)}}{T + \zeta T}$$

$$= \frac{M * E_{(av)}}{T * (1 + \zeta)} \quad (9)$$

If the transmission of the video segment *m* starts d_m seconds before the playback starts and it requires T_m seconds for playback then

$$T_m = \alpha T : 0 < \alpha \leq 1 \quad (10)$$

and

$$d_m = \beta T : \beta \geq 0 \quad (11)$$

where β is a constant.

The average bandwidth $B_{E(av)}$ required to transmit the segment *m* is given by

$$B_{E(av)} = \frac{E_{(av)}}{T_m + d_m} = \frac{E_{(av)}}{\alpha T + \beta T}$$

$$= \frac{E_{(av)}}{T * (\alpha + \beta)} \quad (12)$$

Dividing equation 12 by equation 9

$$\frac{B_{E(av)}}{B_{v\,(av)}} = \frac{E_{(av)} / T(\alpha + \beta)}{M * E_{(av)} / T * (1 + \zeta)}$$

$$= \frac{(1 + \zeta)}{(\alpha + \beta) * M}$$

$$B_{E(av)} = \frac{(1 + \zeta)}{(\alpha + \beta) * M} * B_{v\,(av)} \quad (13)$$

The value α is a constant and it depends on the video playback rate (frames per second). The playback rate depends on the video standard. The constant M is fixed during the segmentation process. Therefore, the average bandwidth required to transmit a video segment

depends on the values ζ and β. If ζ is fixed, the bandwidth required is minimum when β is maximum. Similar relationships can be established between any two segments in the consecutive levels.

If segment m has L_m number of scenes and $C_{m,l}$ represents the size of l^{th} scene of the m^{th} segment in bytes and then the size of segment m can be represented as

$$E_m = \sum_{l=1}^{Lm} C_{m,l} \quad (14)$$

If the scene $C_{m,l}$ has $K_{m,l}$ number of shots and the size of k^{th} shot in this scene is represented as $H_{m,l,k}$, then the size of this scene is represented as

$$C_{m,l} = \sum_{k=1}^{Km,l} H_{m,l,k} \quad (15)$$

If the shot $H_{m,l,k}$, has $J_{m,l,k}$ number of GOPs and the size of the j^{th} GOP in this shot is represented by $G_{m,l,k,j}$, then the size of this shot can be given as

$$H_{m,l,k} = \sum_{j=1}^{Jm,l.k} G_{m,l,k,j} \quad (16)$$

A GOP ($G_{m,l,k,j}$) represents its size in bytes, which is the sum of the bytes of each frame placed between the intracoded frame of that GOP and the intracoded frame provided in the immediate next or previous GOP. The structure of GOP does not depend on the segment. Generally, the structure of GOP depends on the encoding method used. For example in MPEG format GOP has the structure of

{IBBPBBPBBPBB}

But most of the video streaming formats use the structure

{IPPPPPPPPPP}

OR

{I I I I I I I I I I I I}

The number of frames *I* in a GOP is fixed in a video. Therefore, from equations 2, 14,15 and 16, the video V given in equation 1 can be represented in hierarchical form as shown below

$$V = \sum_{m=1}^{M} \sum_{l=1}^{Lm} \sum_{k=1}^{Km,l} \sum_{j=1}^{Jm,l.k} \sum_{i=1}^{I} f_i \quad (17)$$

If the transmission of a video segment, scene, shot or GOP starts *D* seconds before the playback of its first frame starts in the client then the buffer size required at any time t_x is $b(t_x)$ and it is given by

$$b(t_x) = | T(t_x+D) - P(t_x) | \quad (18)$$

Where T is the transmission schedule function at any instance of time t_x and P is the playback schedule at time t_x. If r(t) is the rate of transmission at any instance of time t_x, then the transmission schedule is given by

$$T(t_x + D) = \int_0^{t_x + D} r(t)\, dt \quad (19)$$

If the rate of transmission is assumed to be constant for a video segment, then

$$r(t) = r : 0 \le t \le t_x + D \quad (20)$$

and

$$T(t_x + D) = r \times (t_x + D) \quad (21)$$

The video data, which is going to be played during the interval t_x to t_{x+1} that is f_x, must be available at t_x and any instance of time t, which lies between t_x and t_{x+1}. Hence, the playback schedule up to x^{th} frame is given by

$$P(t_x) = \sum_{x=0}^{i} f_x \quad (22)$$

Where f_x is the number of bytes in the x^{th} frame.

The following simplified algorithm helps to evaluate the streaming parameters such as buffer and bandwidth requirements for each segment, scene, shot and GOP of a streaming video.

```
initializeVariables( )
{
   framePeriod =0.04   // frame rate is 25 fps
   delayBasedTransmission = true
   vBRTransmission = false
   countFrames = 0
   prevDelay = 0
   dDelay = 0     //delta Delay i.e small change in the delay
   newTotalDelay  = 0;
   minDelay = 0
   prevTxRate = 0
   dTR = 1000          //delta transmission Rate
   newTxRate = 0
   maxTxRate = 0
  T = 0                //Transmission Schedule
  P = 0                //Playback Schedule
  maxBuf = 0
  physicalBuffer = 100000 //physical buffer allotted in
                          //bytes
 }

calcResources( segStartFrameNo, segEndFrameNo )
{
 fEnd = segEndFrameNo – segStartFrameNo
 do
 {
countFrames = countFrames + 1
T = newtxRate*((framePeriod* count Frames)
                + newTotalDelay )
P = P + sizeof (countFrames)
bufRequired = T-P
if ((bufRequired>0) && (bufRequired>maxBuf))
 {
  maxBuf = bufRequired;
  }
if (delayBasedTransmission)
  {
     prevDelay  = newTotalDelay  //save the delay value
                                 //already used
     if (bufRequired < 0)  dDelay =
                -bufRequired /  transRate
     if (bufRequired > physicalBuffer)
      {
                          bufRequired - physicalBuffer
        dDelay  =  ------------------------------------
                                 transRate
       }
      newTotalDelay = prevDelay  + dDelay
  }

  if (vBRTransmission)
```

```
    {
      prevTxRate =  newTxRate
      if (bufRequired<0) newTxRate = prevTxRate  + dTR
      if (bufRequired > physicalBuffer)
       {
        newTxRate = prevTxRate  - dTR
       }
     }
  if((newTotalDelay>prevDelay) ||
                          (newTxRate>prevTxRate)
   {
     minDelay = newTotalDelay  //minimum initial delay
                 //required for jitter free transmission
     maxTxRate = newTxRate
     countFrames = countFrames – 1
     }
} while (countFrames < fEnd)

initializeVariables()
calcResources( segStartFrame, segEndFrame )
avBWRequired = P / (countFrames * framePeriod)
Display   avBWRequired, maxBuf, minDelay and
          maxTxRate
```

Implementation and testing: In order to provide an intelligent access to video, an XML meta-data that represents semantic description of the video segment in the hierarchical structure is created. A conversion tool is developed to create this structure. It uses a predefined XML template from a file or database. Initially the user or developer views and listens the streaming video using the play, pause, fast forward and rewind buttons. When the required segment appears, the user describes it by an identifier and attaches the identifier to the starting frame of the segment using connect button. The identifier may be as simple as scene2, shot1, GOP5 etc., or it may be a word or group of words that describe the segment semantically. When the identifier is a semantic description, it can be used for content-based retrieval. The identifiers are the nodes of the XML tree. This tool helps to either insert or delete a node visually in the hierarchical structure. This information is stored in a database. When the summarization of video is required at client side the user has to enter an identifier for which the video content is required. If same identifier has been attached to many parts of the video, the video will be summarized based on its temporal relationships. With a representation frame for each identifier, the user can zoom-in the required part of the video with maximum size and quality by just clicking on it.

The binary tree video structure shown in Fig. 2 does not require any special tool or manual assistance for being created. It can be easily created by segmenting the video based on time indexing. But this is not suitable for content-based retrieval.

The analysis with simulated video frames shows that a video segment requires higher bandwidth than its average bandwidth. But this requirement can come down to some of the segments at lower levels. The average bandwidth required to transmit a video segment depends on the values ζ and β. If ζ is fixed, the bandwidth required is minimum when β is maximum. To avoid jitter, an initial delay can be allowed, but this increases buffer requirement at client side. Actually this initial delay is the sum of the delays at lower level segments, which causes jitter between the segments at lower levels. But higher level segments require larger initial delay and larger buffer to avoid jitter.

Since the streaming parameters such as buffer and bandwidth requirements for each segment, scene, shot and GOP of a streaming video depend on the initial delay, this type of analysis will help the streaming server to create a better transmission schedule such that jitter-free playback is provided at the client. The initial delay may be reduced to a small value if size of the segment is small. In the algorithm shown above, if delay based transmission is chosen, delay is introduced at the beginning of the playback to keep the bandwidth and buffer requirement within the given limits. An increase in initial delay will reduce average bandwidth required to transmit a video segment, but at the same time, it may increase the buffer size required at the client. On the other hand, if VBR transmission is chosen, transmission rate is changed to limit the buffer requirement and to reduce the delay or jitter.

CONCLUSION

The hierarchical summarization implemented in this study requires human assistance for both low-level and high-level segmentation; it can be improved to use automatic shot detection techniques for low-level segmentation and human assistance for high-level segmentation. This will reduce the authoring time for video segmentation. The tool developed for segmenting the video can extend its function to different video archives and the semantic relationship between them can also be established. Generally, content-based retrieval is domain specific, but the method described in this study is not specific to any domain. Finally, the mathematical models developed in this study helps to decide the network resources required to transmit streaming video segments efficiently. Even though simulated video frames have been used to test the algorithm, this algorithm can be included in a video encoder to analyze actual video.

REFERENCES

1. Mills, M., J. Coher and Y.Y, Wong, 1992. A magnifier tool for video data. Proc. ACM Computer Human Interface (CHI), May.
2. Ronald, S., J. Hunter and D. Kosovk, 2004. FilmEd-Collaborative video indexing, annotation and discussion tools over broadband networks. Proc. 10th Intl. Multimedia Modeling Conf. (MMM04) IEEE Computer Society. http:// metadata.net/filmed/pub/MMM04 - FilmEd.pdf.
3. Wallapak, T. and J. Zhou, 2004. Shot clustering techniques for story browsing. IEEE Trans. Multimedia, 6: 517-527.

4. Jia-Yu, P., H. Yang and C. Faloutsos, 2004. MMSS: Multi-modal story-oriented video summarization. Proc. Fourth IEEE Intl. Conf. Data Mining, Computer Society.
5. Dulce, P., S. Srinivasan, A. Amir, D. Petkovic and D. Diklic, 1998. Key to effective video retrieval: Effective cataloging and browsing. ACM Multimedia'98: 99-107.
6. Piera, P., L. Petraglia and G. Petraglia, 2002. The virtual image in streaming video indexing. Proc. Intl. Conf. Dublin Core and meta-data for e-communities, pp: 97-103.
7. Damg, S.Z. and J.F. Nunamaker, 2004. A natural language approach to content-based video indexing and retrieval for interactive E-learning. IEEE Trans. Multimedia, 6: 450-458.
8. Nikolaos, D. and A. Doulamis, 2001. Efficient video transmission over internet based on a hierchical video summarization scheme. IEEE Intl Conf. Multimedia and Expo, Computer Society.
9. Shih-Fu, C., 2003. Content-based video summarization and adaptation for ubiquitous media access. Proc. 12th Intl. Conf. Image Analysis and Processing, IEEE, Computer Society.
10. Manthias, L. and J. Jutta. XML and MPEG-7 for Interactive Annotation and Retrieval using Semantic Meta-data.
11. Chabane, D., 2002. Content-based multimedia indexing and retrieval. IEEE Multimedia, pp: 18-22.
12. Lienhart, R., S. Pfeiffer and W. Effelsberg, 1997. Video abstracting. Commun. ACM, 40: 1531-1541.
13. Smith, M.A. and T. Kannnde, 1997. Video skimming and characterization through the combination of image and language understanding techniques. Proc. Computer Vision and Pattern Recognition, pp: 775-781.
14. Nevenka, D., H.-J. Zhang, B. Shahraraj, I. Sezan, T. Huang and A. Zakhor, 2002. Applications of video-content analysis and retrieval. IEEE Multimedia., April-June: 42-55.
15. Tina, T.Z. and J.S. Jin, 2005. A structured document model for authoring video-based hypermedia. Proc. 11th Intl. Multimedia Modeling Conf. (MMM'05) IEEE, Computer Society.
16. Mufit, A.F. and M. Tekalp, 2003. Two-stage hierarchical video summary extraction to match low-level user browsing preferences. IEEE Trans. on Multimedia, 5: 244-256.
17. Daniel, M.R., 2000. A design pattern- based video summarization technique: Moving from low-level signals to high-level structure. Proc. 33rd Hawaii Intl. Conf. on System Sciences.
18. Yu, F.M. and H.J. Zhang, 2005. Video snapshot: A bird view of video sequence. Proc. 11th Intl. Multimedia Modeling Conf. (MMM'05) 2005 IEEE.