In-Source Video Summarization

Kousik Sankar Ramasubramaniam
Architecture Group,
Cisco Video Technologies Pvt Ltd,
Bangalore, INDIA
kousrama@cisco.com

Ganesankumar Annamalai
Distinguished Engineer,
Cisco Video Technologies Pvt Ltd,
Bangalore, INDIA.
gaannama@cisco.com

Abstract— **State-of-the-art broadcast/transmission standards such as ATSC/DVB/ISDB are based on MPEG as the underlying compression standard. As AV content increases and as newer applications/services – under the Internet-of-Things paradigm -- get enabled, which require different handling of audio/video metadata; it becomes increasingly infeasible to use the traditional video summarization algorithms due to the disparate nature of the AV content and the associated meta-data. This paper presents a mechanism to define and efficiently utilize the meta-data for summarization without any heavy-duty processing on the video encoding / decoding pipeline. For production house content as well as user generated content, the meta-data can be inserted into the stream. The main advantage of this approach is the portability across various platforms and decoder implementations. The overheads on the bit-stream, processing and memory due to such a change are also depicted.**

Keywords-video summarization; in-source video summarization; meta-data

I. Introduction

The state of the art transmission standards are ATSC / DVB / ISDB which are based on MPEG as the compression standard. As the number of transmitted channels and video recordings increase, it becomes increasingly difficult for the end-user to keep track of programs that have been watched and those that haven't been watched. Movie summarization provides a nice way for the end-user to determine whether a movie is worth watching or to determine if he/she has already watched the movie or not [4].

Various users will recall different genre-based scenes of the same movie. Most movie content is of mixed genres (at most three or four different genres, rarely five or more), hence the user will recall very easily his/her most liked genre. For example, in a thriller movie, if the user's recall profile is romance, then the romance scenes will be very easily recallable by the user [1], [2]. There is a high probability that such scenes will be ignored by current movie summarization algorithms, since such algorithms work based on the relative video attributes (eg brightness, contrast etc). Thus, currently, it is impossible for movie summarization algorithms to cater to the entire multitude of users' preferences.

Current movie summarization algorithms, that are video-based, perform the digital de-compression until (and without performing) the Inverse Discrete Cosine Transform (IDCT). This approach works using a pre-allotted CPU/decoding bandwidth. It is also not possible to extend it easily for various genres of movies (suspense, action, romance, thriller, horror, drama) where it is crucial that the movie summarization must not reveal the secretive plot. Hence there is always some form of manual intervention required to verify the correctness of the summarized video output and edit it, if required [4].

This paper proposes a mechanism to embed video summarization information into the source stream itself in appropriate data structures to enable easy decoding in a standard-compliant manner.

II. Source Based Summarization

The current video summarization algorithms are heuristic in nature with various thresholds and analysis techniques to extract summary information [2], [3]. This paper discusses a technique that is not heuristic in nature and attempts to bring consistency in the summarization results across products [5].
There are 3 main problems associated with traditional movie summarization algorithms:-

1. There is no guarantee that the algorithms will work for all genres of video broadcasts (eg news, sports and other genres of movies) To add to the complexity, the algorithms also differ based on the kind of sports! The user's recall will always be based on his most liked genre which is not catered to.
2. The traditional algorithms still require manual intervention to check the correctness of the summarization, though sometimes this is not done due to laborious effort [2], [3].
3. The time taken is far too much especially if it requires partial video decoding, analysis and comparison with other video frames.

Thus, instead of choosing complex decision-based algorithms, there is a strong need to simplify the whole process and make movie summarization easy – almost real-time. This can be achieved if the information is embedded into the source stream. Broadly, there are 2 categories of source streams:-

a. Production Content i.e. content that is produced in the studios and yet to be broadcast.
b. User Content i.e. content that is not broadcast but generated by the user and stored in the cloud or in CE devices at the user's home.

For both types of content, the mechanism of inserting the meta-data into the stream is the same. As content moves from the broadcast server to the network/cloud to the user's device(s), there is no need for any special handling of the meta-data. As long as the content is streamed or stored on the

978-1-4673-8364-6/16/$31.00 ©2016 IEEE

device, the meta-data is guaranteed to be handled properly as long as the video decoder understands the syntax. In the scenario where the video decoder does not understand the syntax, the meta-data will be skipped while playing back the content.

A. Summarization Mechanism

The following highlights the main attributes of our mechanism along with the bit-stream syntax depicted in the figure below:-

- Define various genre entry points within the movie. This will be limited to the sub-genres present in the movie, for the convenience of the user. Based on the user query, the selected genre content will be assembled and played back.
- Define appropriate entry and exit points so that the entire video sequence need not be analyzed every time to extract the summarization information. For the worst case, the sequence analysis needs to be done only one-time for automatic detection scenario. The genre entry and exit points can be defined in three ways:-
  - Automatic detection or
  - User-based entry or
  - Combination of both with the user validating the automatically detected entry points.
- Define various rankings within the video content for each available genre. These rankings are intended to rank the chosen video frames based on their relevance to be chosen as the summary. A higher ranking indicates a higher relevance of the scene to be a representative of the movie.
- Define various time-based thresholds, based on the rankings mentioned above. For e.g., one broadcaster/end-user wants the summary to be 60 seconds long, another might want it to be 120 seconds long etc. With this approach, given the requested summary length, the top N rankings have to be chosen such that the length of the movie summary is equal to the requested length.
- Insert and use the MPEG user data field to convey information about the specific frames that can be used as a representative frame. Define appropriate entry and exit points within the user data.
- For user generated content, use traditional algorithms for summarization to generate the genres / rankings / thresholds / frames within each scene, and provide hooks into the user data fields via which this generated data can be inserted into the stream.

For A/V editing scenarios, where parts of A/V sequences are cut or joined together, modify the entry and exit points in the video user data appropriately to reflect the new edited video summarization.

B. Bitstream Syntax

Figure 1 shows a typical GOP in a MPEG stream consisting of I, P and B frames.

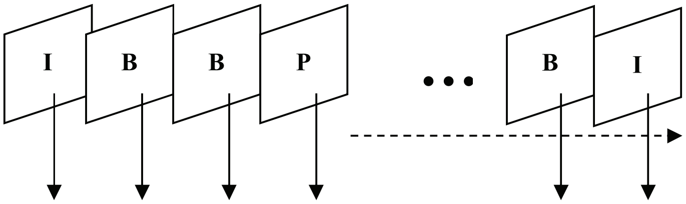

Figure 1: A typical GOP

Figure 2 shows the user data syntax in MPEG that is defined for the summarization information. The MPEG standard defines user data fields which can be transmitted and stored for every GOP or for every field. Two such fields constitute one video frame. Figure 2 defines user data at the GOP level. The logical interpretation of each of the bytes is defined below:-

```
MOV_SUMM_BIT -----------------------------1 bit.
If (MOV_SUMM_BIT==1)
{
   RANK ; -----------------------------------------4 bits
   GENRE ; -------------------------------------32 bits
    {
       FWD_DIR_EXIT_COND ; -------------1 bit /* exit to GOP
pointed to by NXT_GOP_FWD_LNK or auto exit to
                              next GOP */
        If (FWD_DIR_EXIT_COND == 1)
        {
          NXT_GOP_FWD_LNK ;  ----------33 bits. /* for forward
playback/scan modes. Not true for last GOP. */
        }
        Else
        {/* nothing to be done, just auto exit to the next GOP after
playback of the current GOP is over. */
        }
    }
    {
         BWD_DIR_EXIT_COND ; ----------1 bit /* exit to GOP
pointed to by NXT_GOP_BWD_LNK or
                                     auto exit to next GOP */
         If (BWD_DIR_EXIT_COND == 1)
         {
               NXT_GOP_BWD_LNK ; ------33 bits. /* for reverse
playback/scan modes. Not true for first GOP. */
         }
         Else
         {/* nothing to be done, just auto exit to the previous GOP
after playback of the current GOP is over. */
         }
    }
}
else (MOV_SUMM_BIT == 0) /* true only for first GOP in the
video sequence*/
{
      NXT_GOP_FWD_LNK ; ----------------33 bits. /* PTS of first
GOP */
      NXT_GOP_BWD_LNK ; ---------------33 bits. /* PTS of last
GOP */
}
```

Figure 2 : User data structure for summarization

978-1-4673-8364-6/16/$31.00 ©2016 IEEE

The block shown above indicates the byte structure within the user data block. The logical interpretation of each of the bytes is explained below:-

MOV_SUMM_BIT : Values 0 or 1. Indicates whether this GOP will be used for summarization or not.

RANK : Values range from 1 to 5. Indicates the ranking of the current GOP from 1 to 5 based on the relevance for the movie summarization. Between 2 GOPs (A and B), if A is ranked 4 and B is ranked 5 and the length of the summarized content is required to be 60 seconds, then B is chosen first and if there are no other GOPs with ranking=5 and the total length of the summarized content is less than 60 seconds, then and only then A is chosen.

GENRE : 32 bits defined by the broadcaster or an option chosen by the user.

FWD_DIR_EXIT_COND and BWD_DIR_EXIT_COND: Values 0 or 1. A ‘1’ indicates “Exit to the next GOP pointed to by NXT_GOP_FWD_LNK or NXT_GOP_BWD_LNK”. A ‘0’ indicates “Continue to the consecutive frame”.

NXT_GOP_FWD_LNK : Indicates the PTS (Presentation Time Stamp) of the next GOP to jump to in the forward direction.

NXT_GOP_BWD_LNK : Indicates the PTS (Presentation Time Stamp) of the next GOP to jump to in the backward direction.

C. *Overhead Considerations:-*

Bitrate/Processing -- The total number of bits per *representative* GOP is about 105 bits of user data. Compared with the *average* transfer rate for digital AV (6 Mbps), the overhead (assuming worst case of 3 GOPs per second) would be about (3 X 105 X 100) / (6 X 10^6) = 0.0052%. All video decoders can easily handle this extra payload.

Memory – This mechanism does not require any extra memory from the processing end apart from the user data memory highlighted above. During browsing of video content, for a smooth navigation experience, it is better to cache the summarization related user data into another data structure to improve performance. This memory is of the order of 3 to 10 KB.

III. Results and Conclusion

Using the user data fields within the MPEG stream, we have derived a mechanism to embed the AV summarization information thereby ensuring portability of the summarization information across hardware platforms and operating systems. The overhead due to such a mechanism is also limited – the bitrate processing is about 0.0052% and the extra memory is about 3 to 10 KB. Future work involves embedding this information into a product and testing/fine-tuning for broadcaster and end-user feedback.

Acknowledgments

We are very thankful to the management at Cisco Video Technologies (in particular Rashma Raveendra and Vladi Ruppo) for their encouragement and support.

References

[1] Xiang Wang, Jing Chen and Caiyun Zhu, “User-Specific Video Summarization”, 2011 IEEE International Conference on Multimedia and Signal Processing (CMSP), Guilin, Guangxi.

[2] Muhammad Ajmal and Faiz Ali Shah, “Video Summarization: Techniques and Classification” Computer Vision and Graphics, Lecture Notes in Computer Science Volume 7594, 2012, pp 1-13.

[3] Wen-Nung Lie, Kuo-Chiang Hsu, “Video Summarization Based on Semantic Feature Analysis and User Preference”, IEEE International Conference on Sensor Networks, Ubiquitous and Trustworthy Computing, 2008. SUTC '08.

[4] Russell, D.M, “A design pattern-based video summarization technique: moving from low-level signals to high-level structure”, Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, 2000.

[5] Source video based Video summarization, 3322/CHE/2013, Patent Filed.

978-1-4673-8364-6/16/$31.00 ©2016 IEEE