SEMANTIC VIDEO SUMMARIZATION IN COMPRESSED DOMAIN MPEG VIDEO

Jek Charlson So Yu, Mohan S. Kankanhalli*, Philippe Mulhem***

School of Computing*/IPAL-CNRS**, National University of Singapore, Singapore 117543
E-mail: jekcharlsonyu@yahoo.com, { mohan, mulhem}@comp.nus.edu.sg

ABSTRACT

In this paper, we present a semantic summarization algorithm that interfaces with the metadata and that works in compressed domain, in particular MPEG-1 and MPEG-2 videos. In enabling a summarization algorithm through high-level semantic content, we try to address two major problems: First, we present the facility provided in the DVA system that allows the semi-automatic creation of this metadata. Second, we address the main point of this system which is the utilization of this metadata to filter out the frames, creating an abstract of a video based on a Boolean condition set by the user. Our video summary quality survey indicates that the proposed method performs satisfactorily.

1. INTRODUCTION

One important utility that sprouted from the field of digital video processing is the concept of video summarization. A video abstract, as the name suggests, "may be defined as a sequence of a still or moving images presenting the content of a video in such a way that the respective target groups is rapidly provided with concise information about the content while the essential message of the original is preserved" [4].

Most of the past research efforts focus on utilizing low-level features as summarization criteria. However, more recent research deal with higher level of inference such as in [2], where higher-level contents mainly action, close-up, crowd, and setting are inferred by processing low-level information particularly skin color, motion intensity, and texture. Others such as CMU Informedia [7] use sophisticated tools to extract semantic information and use this data to produce automated video summaries. Despite these advancements, there are certain drawbacks with these systems since it does not provide functionality for customization. These systems simply output video summaries based on heuristics and assumptions, and may not be suited for the user needs.

The semantic summarization algorithm in this paper proposes another approach by considering a two-step process: the high-level stage to identify the relevant frames based on a user-specified condition and further processing will involve using low-level processing to meet the desired length. The summarization is done in compressed domain since the processing technique is comparable to the uncompressed domain and yet holds an advantage because of high speed.

In this paper we begin by giving an overview of the system as well as the low-level feature extraction and processing adapted in the system. Afterwards, the semantic summarization algorithm and results of the experiments will be expatiated.

2. DIGITAL VIDEO ALBUM

The Digital Video Album (DVA) project aims to develop techniques for content-based indexing, intuitive access and retrieval of digital images and video. DVA also aims to develop methods to index, mine, summarize and access digital video sources such as digital TV. It consists of 6 modules and 3 of which are tools providing data as shown in figure 1.

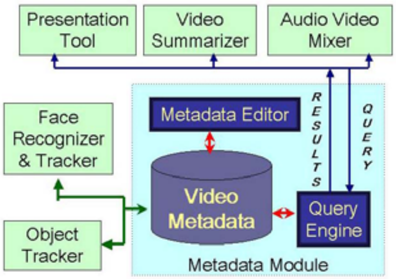

Figure 1: DVA System Architecture

The object and face tracker are automatic annotation tools that detect and track objects and faces in the video and store this information in the XML metadata, the metadata editor allows manual input to the XML Metadata, and finally the XML metadata provides the data needed by the semantic summarizer. Access to the database is made possible by the query engine.

It is important to note that the metadata is an integral part of the DVA system because it serves as the database and coordinates the information of the whole system. Thus the user is not required to manually annotate every frame with respect to an object or person, but rather the DVA has the capability to expedite the creation of such metadata. The video summarization simply takes advantage of the fact that the system already has a metadata. For more information on DVA, please refer to [9]

3. EXTENDED BOOLEAN ALGEBRA [5]

In identifying the frames that will be included in the video summary, we adapted the extended Boolean Algebra model since it has combined the advantages of different models, the structuredness of Boolean Algebra conditions and the ability to rank documents in probabilistic and vector-processing models.

The Extended Boolean Algebra considers two (or more) term-queries either (A *or* B) or (A *and* B). The *and* operation aims to satisfy both terms whereas the *or* operation avoids a situation where both terms are not satisfied thus it is evident that for *and*-queries, the (1,1) point is the desirable location whereas for *or*-queries, the (0,0) point is the undesirable location. The following similarity functions may be derived based on the Euclidean distance for *or* and *and*:

$$sim(D, Q_{(A\,or\,B)}) = \sqrt{\frac{d^2{}_A + d^2{}_B}{2}} \tag{1}$$

$$sim(D, Q_{(A\,and\,B)}) = 1 - \sqrt{\frac{(1 - d^2{}_A) + (1 - d^2{}_B)}{2}} \tag{2}$$

where : D : Document

$Q_{(A\ X\ B)}$: Query A X B; X = *and* or *or* operator

d_A : weight of term A; d_B : weight of term B

This similarity function is then adapted to our algorithm, and we modify it by considering frames as documents since each frame has certain information attached to it and the video as the collections of all the documents. To elaborate, given that our Video V = frame {1, 2, 3, 4}, and A and B are objects or people identified in the system. The table below gives the particular semantic value for each frame.

Frame #	Terms		Similarity function	
	A	B	A or B	A and B
Frame 1	1	1	1	1
Frame 2	1	0	1/ √ 2	1 – 1 / √ 2
Frame 3	0	1	1 / √ 2	1 – 1 / √ 2
Frame 4	0	0	0	0

Table 1: Extended Boolean Algebra Similarity Function

In addition to the *and* and *or* operators, we also have to consider the unary operator *not*. The *not* operation is evaluated as follows as *not* (A) = 1 – d_A. Thus frames which have values of 1 and .70 will have 0 and .30 respectively after the *not* operation.

4. LOW-LEVEL PROCESSING

The semantic summarization algorithm involves a two-step process where semantic contents are used in the first stage of processing and low-level features are used for further processing. In this section, we will be discussing the different strategies and techniques utilized in the algorithm for extracting and handling low-level features.

4.1. Feature Extraction – Color Information

In extracting color information in compressed domain, a feature known as DC image is used [8]. The DC image provides a good indication of the information of the compressed video however due to the storage problem as well as the inefficiency of the computations of DC Image, a DC histogram is used to save the feature of the frames of the specified video. For the sake of efficiency and reduced dimensionality, we will only consider luminance blocks when forming the DC image. This is because the eye is sensitive to small changes in luminance, but not in chrominance. Thus we can discard the chrominance information without affecting the quality of the extracted DC image much. In this paper, we will simply refer to luminance as the *color information*.

The histogram will be divided into 64 bins and each bin will account for 4 luminance values. Since luminance values ranges from 0 (black) to 127 (gray) to 255 (white), it can be assumed that this range is linear and thus values close together are similar. To calculate the difference between two histograms, the sum of the absolute bin-to-bin difference is taken.

4.2. Feature Extraction – Motion Information

In calculating motion vectors, we consider the difference in inferring P, B, and I frames of the MPEG video. I frames' values are set to 0 since they are intra-coded. The P frames have a straightforward solution. However, in B frame, a consideration of the two motion vectors must be taken into account. Therefore, an average of both forward and backward motion vectors B = |A + C| /2 is considered. Furthermore, in determining the motion intensity of the motion vectors, we have adapted the variance motion intensity and may be computed as follows [8]:

$$avg = \frac{1}{N}\sum_{i=1}^{N}\left\|\vec{V}_i\right\| \ ; var = \frac{1}{N}\sum_{i=1}^{N}\left\|\vec{V}_i\right\|^2 - avg \tag{3}$$

$\vec{V}_i$: Motion vectors in the frame $(i = 1..N)$

N : Number of motion vectors in the frame

4.3. Color Similarity Expansion

The color similarity expansion utilizes the color information of each frame to include frames based on how similar the frame is to the accepted frames adjacent to it. Accepted frames are frames which have already been identified as frames to be included in the video summary. The algorithm is shown below:

```
Given   N': number of frames desired
        V: Video, containing 1 to N frames
Ct = current number of accepted frames,
While (Ct < N') {
        Initialize all frames except accepted frames to 0;
        For (I = 2; I < N; I++) {
        If V[I] = 1  // it is an accepted frame
                V [ I – 1] = absolute difference of Color
                histogram of V [I] and V[I – 1|
                V [I + 1] = absolute difference of Color
                histogram of V [I] and V[I + 1]}
        Search frame with lowest color difference and set it's
        value to 1 (accepted frame).
        Ct=Ct+1; }
```

4.4. Content-Based Adaptive Clustering Algorithm [6]

The Content-Based Adaptive Clustering (CBAC) algorithm is illustrated in figure 2. To expatiate the concept of CBAC, consider a Video *V* with an *N* number of frames and *N'* as the desired number of frames. The system first extracts the low-level feature required for the summarization such as motion intensity and color information as discussed in section 4.1 and 4.2. Parameters *L* and *r* (0<r<1) are then initialized. Clustering is made by grouping *L* number of adjacent frames as a unit. To elaborate, given that there are 9 (numbered 0 – 8) frames in the video, the video will be partitioned as follows given *L* = 3: {(0,1,2), (2,3,4), (4,5,6), (6,7,8)}. Unit change for each cluster is computed from the first and last frame of the unit. It is important to note here that the unit change may be based from the low-level content features such as color and motion.

The unit changes forms an array of length $K=\lceil N/(L-1) \rceil$ and are arranged in ascending order. The initialized parameter *r*, specified by the user, categorizes the clusters into two categories: *small-change* and *large-change clusters*. The small-change clusters are those in *K*r* of the array, and frames in these are all removed except for the first and last frames of each unit. After the deletion process, *K*r*(L-2)* are deleted.

Given that the number of frames left is *N"*, if *N" <= N'* then the desired result is achieved. On the other hand, if not, then the frames retained are then regrouped and the clustering algorithm is then repeated. The iteration continues until the desired number of frames is achieved. The frames that are not discarded are considered the representative frames (R-frames).

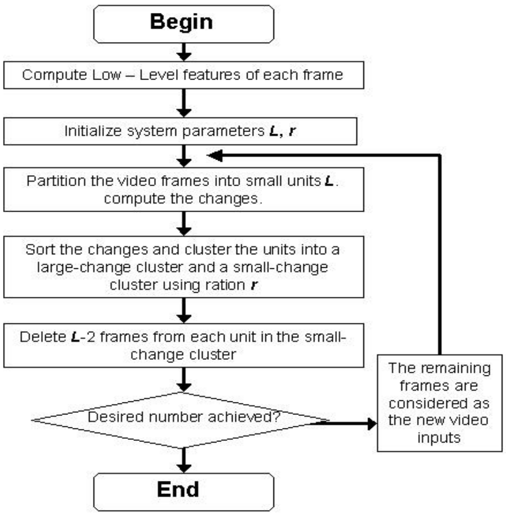
Begin
Compute Low – Level features of each frame
Initialize system parameters L, r
Partition the video frames into small units L. compute the changes.
Sort the changes and cluster the units into a large-change cluster and a small-change cluster using ration r
Delete L-2 frames from each unit in the small-change cluster
Desired number achieved?
The remaining frames are considered as the new video inputs
End

Figure 2: Content-Based Adaptive Clustering Algorithm

5. SEMANTIC SUMMARIZATION

Figure 3 illustrates the semantic summarization algorithm. The first stage involves a computation of the semantic values of the system wherein the input is a user-specified Boolean condition specified. This is where our system mainly differs from other video summarizers since it allows customization by allowing users to specify which particular object, people, or video sequences is most important and should be included in the video summary. The system provides two interfaces for specifying a condition: A simple interface that provides list of objects, names, and other information, and allows user to click and add terms to the condition; and an advanced interface where user has to specify the Boolean condition himself. A computation of the semantic values by evaluating the Boolean condition thru the Extended Boolean Model is made and the values are assigned to their respective frames. A frame count is then done to determine how many frames fully ($\mathbf{d}_X = 1$) or partially ($\mathbf{d}_X < 1$ and $\mathbf{d}_X > 0$) satisfy the given condition and based on these, three possible situations are observed. Consider *S* as the number of frames partially or fully satisfying the condition and *N'* as number of frames desired. The three cases are S = N', S < N', and S > N'.

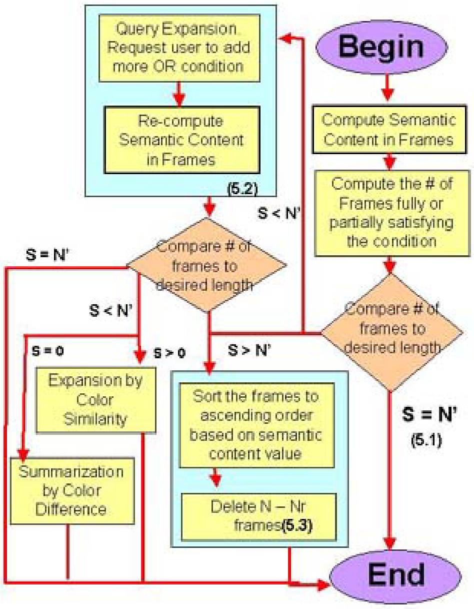
Query Expansion Request user to add more OR condition
Re-compute Semantic Content in Frames
(5.2)
Begin
Compute Semantic Content in Frames
Compute the # of Frames fully or partially satisfying the condition
S < N'
S = N'
Compare # of frames to desired length
S < N'
Compare # of frames to desired length
S = 0
S > 0
S > N'
Expansion by Color Similarity
Sort the frames to ascending order based on semantic content value
S = N'
(5.1)
Summarization by Color Difference
Delete N – Nr frames(5.3)
End

Figure 3: Semantic Summarization Algorithm

5.1. When Desired Length is achieved

The first case is when the number of frames satisfying the condition is equal to the number of frames desired for summary (*S = N'*). In this situation, the system simply discards the frames that does not satisfy the condition. This is a straightforward solutions and thereby does not involve further processing.

5.2. When Desired Length is greater than the Number of Non-zero Valued Frames

There are two resolutions possible when S < N': one is to make certain assumption of needs of the user and automatically include frames; and the other is to allow a user feedback option. Our system combines both methodologies by first asking the user for feedback known as the *Query Expansion Stage* and if further processing is still needed, then an automatic low-level processing is done. The Query Expansion Stage involves asking the user to add more terms in the condition and adding it to the original condition with an *or* operator. A simple example is when a N = 100 and N' = 80. If the given condition "*John and Jane*" returns only 40 frames then the system prompts the user to expand the condition to meet the desired length and may expand it as "John *and* Jane *or* Burning House *or* Cutting of Cake."

Afterwards, frames are re-computed and new values are assigned to the frames. If the output is still S < N', another approach, either summarization by color difference or color similarity processing, is considered. If the number of frames retrieved based on query expansion is zero then ordinary summarization of CBAC (see section 4.4) using color information is used, however if it is more than zero then the current retrieved frames is expanded using color similarity expansion. This is based on the assumption that adjacent frames are usually part of the video sequence, and thus frames to be included should be similar to the already chosen frames.

5.3. When Desired Length is less than the Number of Non-zero Valued Frames

In the case of S > N', there are two possible solutions similar to 5.2: feedback or automatic processing. Our system adapted a faster solution that uses the motion intensity criterion to reduce the number of frames. Thus, in our system, we first sort out the frames and frames identified as *border-valued* frames will undergo CBAC (see section 4.4) using motion intensity. To illustrate the concept of border-valued frames: given N' = 3 and V = {1_F = 1, 2_F = .80, 3_F = .5, 4_F = .5, 3_F = .1, 6_F = .2}, where 1_F = 1 is Frame 1 with weight or value of 1. In this case, 3_F and 4_F are border-valued frames.

6. EXPERIMENTS AND RESULTS

	Video Genre	Duration	Summary Ratio
A	News	5m 01s	10%
B	Home	4m 41s	20%
C	Commercial	0m 30s	50%

Table 2: Videos used in the experiment

Video	Clarity	Conciseness	Coherence	Overall Qual.
Predefined Video Summaries				
A	4.6	4.8	4.1	4.4
B	5.3	5.6	4.8	5.2
C	5	5	5	4.9
User-Defined Video Summaries				
	5.3	5.4	4.9	5.4

Table 3: Results of Video Summary Quality User Survey

Usability Survey	
I. Overall Reaction to the software	6.73
II. Screen	7.23
III. Terminology and System Information	6.58
IV. Learning	6.90
V. System Capabilities	6.75

Table 4: Results of Usability Survey

We have conducted a quality and usability assessment survey involving 20 participants, mostly students and using three types of videos shown in table 2. We have three video sets and created predefined video summaries as well as allowed the user to create their own video summaries and rate the results on their perception of how the video should be summarized based on the semantic condition set. The quality of the video summaries is determined by *Clarity, Conciseness,* and *Coherence.* It can be inferred as shown in table 3 that the participants perceive the summary produced by the system as adequate in capturing the essential information of the video since generally the videos get a rating of more than 4, which a good average for a range of 1 to 7, with 7 as the highest rating. As for the Usability survey, we have adapted the Questionnaire for User Interface (QUIS)[1]. There are five categories, which can be rated from 1 to 9, with 9 being the highest. Based on the results shown below in table 4, it may be inferred that the participants find the system user-friendly and usable. For more details, please refer to [9].

7. CONCLUSION

The DVA system is an integration of various systems such as the face recognizer, object tracker, and metadata module that serves as annotation tools of the video. The DVA semantic summarizer uses the information produced by these tools in determining the semantic values of the frames and Extended Boolean Algebra is used to evaluate the condition. Further processing of the summaries are made through low-level processing such as the Content-Based Adaptive Clustering and the Color Similarity Expansion to meet the desired length for the video summary.

Experiments conducted show that the system produces reasonable video summaries. However, it is still evident that there's a limitation to the current system since the video summarizer is dependent on the metadata. Therefore, the quality of the summarization depends mainly on the richness of the metadata of the video.

Future work would involve giving weights to each term specified in condition and providing richer metadata by involving more sophisticated annotation tools.

8. REFERENCES

[1] J.P. Chin, V. A. Diehl. and K. L. Norman, "Development of a Tool Measuring User Satisfaction of the Human-Computer Interface," *Proceedings of SIGCHI '88.* pp. 213 – 218, New York, 1988.

[2] J. Nam and A. H. Tewfik, "Dynamic Video Summarization and Visualization," *ACM Multimedia '99.* (Part 2), pp. 53 – 56, Bristol, 1999.

[3] K. Peker. A. Divakaran and T. Papathomas, "Automatic measurement of intensity of motion activity of video segments," *SPIE Storage and Retrieval for Media Databases* 2001, pp 341 – 351, Santa Clara, January 2001.

[4] S. Pfeiffer, R. Lienhart, S. Fischer, and W. Effelsberg, "Abstracting Digital Movies Automatically," *Journal of Visual Communication and Image Representation,* Vol. 7, No. 4, pp.345 – 353, December 1996.

[5] G. Salton, E. A. Fox. and H. Wu, "Extended Boolean Information Retrieval," *Communications of the ACM,* Volume 26, Issue 11., pp. 1022 – 1036, November 1983**.**

[6] X. D. Sun and M. S. Kankanhalli. "Video Summarization Using R-Sequences," *Journal of Real Time Imaging,* Vol. 6, No. 6, pp. 449 – 459, December 2000.

[7] H. Wactlar and M. Christel, "Digital Video Archives: Managing Through Metadata." In *Building a National Strategy for Digital Preservation: Issues in Digital Media Archiving,* Commissioned for and sponsored by the National Digital Information Infrastructure and Preservation Program, Library of Congress, April 2002.

[8] B. L. Yeo and B. Liu, "Rapid Scene Analysis on Compressed Video," *IEEE Transactions on Circuits and Systems for Video Technology***,** Vol. 5, No. 6, pp. 533 – 544, December 1997.

[9] J. C. S. Yu, "Semantic Video Summarization in Compressed Domain MPEG Video," *M.S. Thesis***,** School of Computing, National University of Singapore. http://diva.comp.nus.edu.sg/ summarizer.htm , July 2002.