Content-Based Representative Frame Extraction For Digital Video

Xinding Sun, Mohan S. Kankanhalli, Yongwei Zhu, Jiankang Wu
RWC* Novel Function ISS Lab
Institute of Systems Science
National University of Singapore
Kent Ridge, Singapore 119597
{xdsun, mohan,ywzhu,jiankang }@iss.nus.sg

ABSTRACT

We present a novel methodology for the extraction of representative frames of a digital video sequence. The proposed method is called content-based adaptive clustering(*CBAC*) which allows a user to focus on his interest in the video using these frames. It achieves this by allowing a user to select the preferred low-level content and the fraction of the frames he would like to extract from a video.

In our algorithm, shot boundary detection is not needed. Video frames are treated as points in the multi-dimensional feature space corresponding to a low-level content such as color, motion, shape and texture. The changes of their distance are compared globally for extraction of representative frames. The frames of the video are dynamically clustered into two clusters according to their changes of distance. One cluster is designated for deletion and the other one is for retention. The algorithm converges to the result desired by the user by deleting some frames from the deletion cluster during each iteration.

Based on our proposed *CBAC* method, we have developed a video player which has the functions of content-based browsing and content-based video summary. While it provides a flexible tool for video review, it can also be a sound basis for other work such as clustering of similar sequences and video retrieval.

KEY WORDS: *digital video, adaptive clustering, representative frame, content-based, video summary, video browsing.*

1. INTRODUCTION

Among the information media delivered through the Internet, digital video is playing an increasingly important role. In recent years, the development of compression and network technology has enabled the creation of a large amount of digital video content. For example, *realplay*, created by Progressive Networks Inc.[7], can disseminate digital videos by using a real time streaming protocol. Owing to the rapid increase in the size of digital video databases, users are being provided with a very broad selection space, and thus are driven to require more flexible as well as powerful video handling tools. Therefore, development of advanced video data management tools is a very important area of research.

Conventionally, people would use a shot-scene structure to describe a video. Defined in Bloedow[4]: A shot is the basic unit of film structure which is a scene from camera start to stop or the length of film from splice to splice in an edited movie. A scene is a unit of one event, location or dramatic incident and thus can be composed of several shots, or only one. However, such a structure is film production oriented. If we want to get detailed information from a video, shot should not be the unit since normally a lot of changes will take place in it. For example, if we want to get a summary of a video, changes within a shot should also be considered.

If we want to concentrate on the change within a shot, we in fact want to extract some important frames to describe it. Here, we define such important frames as *representative frames*. Even now, automatic understanding of image and video content in terms of semantics is not possible. So video processing is still based on low-level content like color, motion, shape,

---

* Real World Computing Partnership, Japan.

texture etc. At the shot boundary, almost all kinds of content of the frames change greatly, so we can apply different metrics to detect it. Within a shot, however, different content of frames may change in a different manner. For example, in a soccer game, the most often dynamically changing low-level content is motion rather than color. So, when we are working on finding representative frames, we should give the result based on a typical selected content. Hence, the above requirement can be formally described as:

Given:

1. an ordered set of input digital video sequence V with cardinality N. $V = \{ F_1, F_2, \ldots, F_N\}$, where $F_1, F_2, \ldots, F_N$ are the frames of V.
2. ratio α such that $0 < \alpha < 1$.
3. low-level content P of {color, motion, ...}.

To extract:
a set of output frames V' with cardinality of N'.

$$V' = \{R_{p1}, R_{p2}, \ldots R_{pN'}\} \quad (1)$$

where

- $N' = N * \alpha$.
- $R_{p1}, R_{p2}, \ldots, R_{pN'} \in V$, are the representative frames of V with respect to feature P.
- $V' \subseteq V$.

So, basically given a digital video V having N frames, we would like to extract a $N*\alpha$ cardinality subset of frames which best represent the content P of the video V.

2. RELATED WORK

An example of early work on video content analysis was carried out by Connor[8]. Specific object changes are detected for key frames. After the development of shot detection techniques such as by Zhang et al.[15], Zabih et al.[14], researchers would select one frame(normally the first frame) from a shot to represent the entire shot. Boreczky et al.[5] compared such shot boundary detection techniques as pixel differences, statistical differences, compression differences, edge track tracking etc. As we have mentioned, to use the shot as the basic unit is not enough for a detailed video analysis, so other researchers have focused their work on finding representative frames.

Zhang et al.[16] have proposed selecting representative frames of different densities by adjusting threshold values. In each shot, the first frame is used both as a reference frame and as a representative frame in the beginning. The distances to this frame for the subsequent frames are computed. When the distance exceeds a given threshold, a new representative frame is claimed and the claimed frame serves as a new reference frame for the following frames. The selected frames can be gathered together to output a sequence.

Yeung et al.[12] have applied a technique similar to that of Zhang's into extracting representative frames for the clustering of video shots. The representative frames selected in each shot serve as members for comparison of different shots. Moreover, Ardizzone et al.[2] have utilized this method for the video database indexing and retrieval.

Smith et al.[10] have proposed a method of selecting key frames based on audio and image. The audio track is created based on keywords. The image frames are selected using a ranking system which regards faces or text as most important, static frames following camera motion the second important, etc. Though the integration of speech, language and image information is the best way to understand a video, the generation of such technique still requires manual intervention and much room remains for improvement.

Detection of shot comes naturally from the video production process but it is difficult to accurately detect all shot boundaries in a video. While Hampapur et al.[6] proposed using post production editing frames to detect shot boundary, advanced video editing techniques can blur the shot boundaries as well. Hence, our work will address on effectively extracting representative frames without detecting shot boundaries.

To meet the above requirement, we propose the *CBAC* technique in this paper. Different fractions of a video possessing different number of frames can be pre-computed according to different low-level content. When required, they can be retrieved and utilized for diverse applications.

3. EXTRACTION OF REPRESENTATIVE FRAMES

When a user is asked to select some important frames according to a typical low-level content like color, motion, shape, texture, etc. he will probably make the choice by the amount of change of the low-level content. In the places where there are more changes he will choose more frames and less number where there is less content change. In fact, the work of extracting representative frames is based on such an assumption.

3.1 THE SPATIAL FEATURE OF VIDEO FRAMES

If we use an M-dimensional feature to analyze a video, all the frames are points in the same Euclidean space E_M. The similarity of the frames is well reflected by the distances of points in the space. A video begins from a point which is the first frame and the temporal sequence of frames traces a path in the space. When the content of frames changes quickly, the path also moves quickly with large steps.

A simple approach to analyze the change of content of frame is to select a fixed reference frame. The distances of all the frames to this reference are computed for comparison with respect to a typical content. The reference frame maybe the first frame of a video, a typical frame, etc. This will map a video into a one-dimensional discrete signal.

Zhang et al. [15] have successfully applied gray-level histogram analysis for shot detection. The color pictures are transformed into gray-level using the National Television System Committee (NTSC) conversion formula:

$$I = 0.299R + 0.587G + 0.114B \tag{2}$$

where R, G, B stands for the intensities of red, green and blue components of a color picture. The city-block distance of two frames X and Y can then be obtained:

$$D_c(X,Y) = \sum_{i=0}^{M-1} |X_i - Y_i| \tag{3}$$

where M is the dimension or number of gray levels here, X_i and Y_i denotes the histogram value of X and Y respectively.

Figure 5.a shows such a mapping result of our test video "news" using histogram as the feature. Distances of all the frames to the first frame are computed, which can reflect the content changes to some degree. However, this measurement does not always work. Suppose X is the first frame of a video and we use it as the reference frame, Y is a randomly selected frame from the same video. Given a constant ω, if the video moves on a hyper-cube:

$$\sum_{i=0}^{M-1} |X_i - Y_i| - \omega = 0 \tag{4}$$

then, such distance measurement can not detect any change.

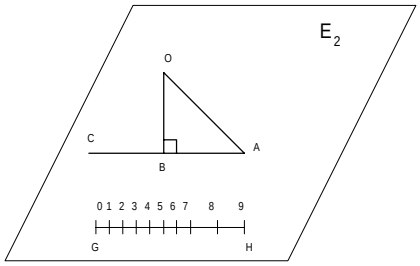


Figure 1. Examples of Video Trajectories in the Feature-Space. $M=2$, $E_M \rightarrow E_2$.

For the sake of simplicity, in figure 1 we assume M=2 and O, A, B,C are frames of a video. So all the points are located on the same plane. If we use O as a reference point, because OB⊥AC, on the line ABC we can not detect any distance change. Actually, we can see that the distance between A and C is even larger than that of OA.

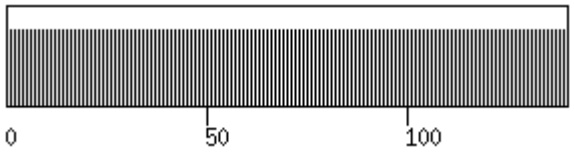
Figure 2 shows an example of such problem. A test video "skate" starts with a black picture whose histogram is $X = (S, 0, \ldots 0)$, where S is the number of pixels of each frame. If we use M-bin histogram for analysis. Then according to (3) we can obtain:

$$D_c(X,Y) = \sum_{i=0}^{M-1} |X_i - Y_i| = 2S - 2Y_0 \tag{5}$$

This will reflect only changes of the first bin of each frame. In the beginning more than 100 frames, Y_0 is very similar, so $D_c(X,Y)$ will not detect change although the real content or value of other bins may change a lot.

Similarly, we can also apply this analysis on Euclidean distance measurement(the problem occurs

when a video moves on a hyper-sphere) and other metrics like χ^2-test[18]. Although the degree of above problem may vary with the metric selected, as a video is generally very long, the overall possibility of the problem occurring is still significant.



The Distances to the First Frame. (a)


Frame 0 and frame 60  (b)

Figure 2. Part of Experiment Result on "skate".

In Zhang[16], the first frame of each shot is used as the reference frame in the beginning and its distances to the following frames are computed. When the distance exceeds a given threshold, a representative frame is claimed and the claimed frame serves as a new reference frame. By changing the reference frame intermittently, the errors due to the above problem may be limited to a small range. However, the above problem still persists. In the case of video summary, we will require a very small fraction of frames to be extracted and hence will require a big threshold. So it is very possible to introduce large errors.

Also, if we want to extract a specific fixed number of frames from a video, then it will be very difficult to control the output number by controlling the threshold. Therefore, we would like to pursue other solutions which can avoid these difficulties.

3.2 USING CLUSTERING METHOD FOR EXTRACTION OF REPRESENTATIVE FRAMES.

As described earlier, the temporal sequence of frames of a video traces a trajectory of points in the content feature space. The nature of the spatial distribution of points of a video can be described as clusters connected by abrupt or gradual changes. During most of the time, the video will move around in a small cluster. It is impossible for all the frames in a video to be spatially far apart and thus have unrelated content, because the frames of a video work together to convey meaningful information. This nature of the distribution of points provides a sound basis for our clustering technique.

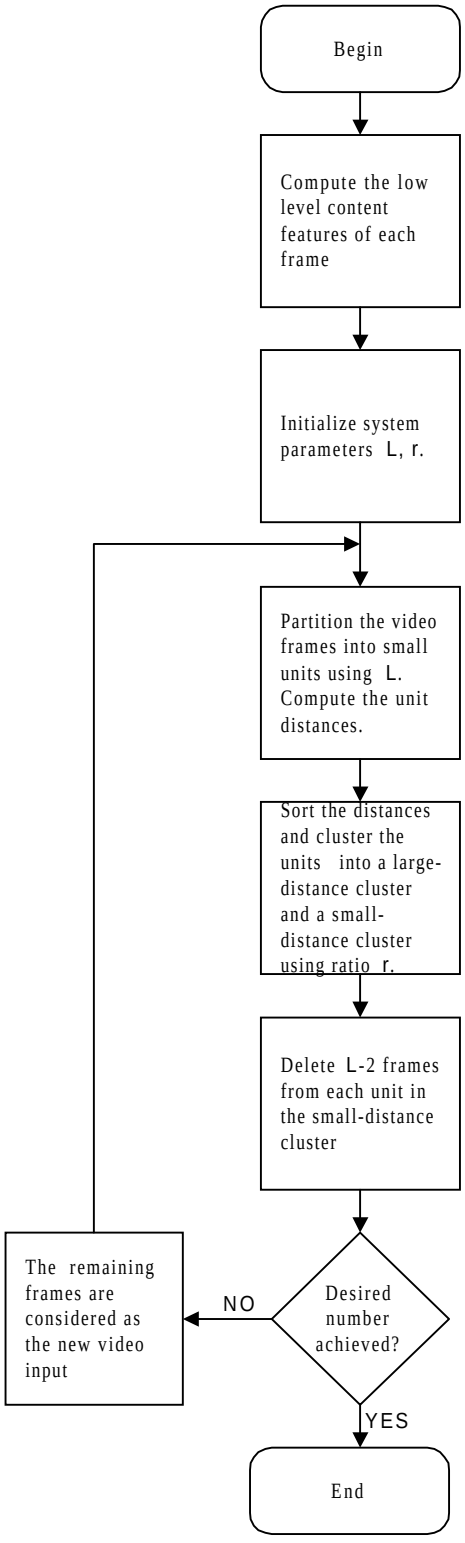


Figure 3. Adaptive Clustering Algorithm.

3.2.1 The Adaptive Extraction Process

The successful use of histogram analysis in the work of Zhang et al.[15] shows that it is a very effective way for video content analysis. Therefore, in our initial work, we also use this technique. For the sake of computational efficiency, we use 64 bins in our work, so all the frames are in the same Euclidean space E_{64}.

Here we give the description based on histogram analysis. Since this adaptive extraction method is a general approach, it is relatively straightforward for us to apply this technique for other content such as color, motion, shape, texture etc.

For a given video V with length N, suppose we want to extract N' representative frames. The histogram of each frame in V is computed first. This algorithm works in an iterative fashion. We start initially with all the frames of the video and iteratively drop frames till the desired result is obtained.

The sequence of the video frames is partitioned into small units whose length are all L. All the units are temporally consecutive. Figure 4 shows the partitioning with L=2 and L=3 respectively. The partitions for L=3 are {(0,1,2), (2,3,4), (4,5,6), (6,7,8)}.In each *partition* the distance called *unit distance* is computed, which is the distance between the first frame and the last frame of the unit.

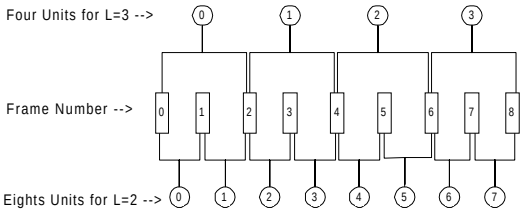


Figure 4. Sequence Partitions.

The computed distances stand for each unit and they construct an array of length $K=\lceil N/(L-1) \rceil$. Because our objective is to extract representative frames according to frame content changes, the distances do reflect the actual degree of content change in all the units. This is because the distance metric is computed in a temporally localized region. By sorting the unit distances in an ascending manner, we get an array which represents the general content change of the video. The elements which are located in the beginning part of the array represent the frames where there are small changes, while the units in the later part consists of frames having large changes.

By selecting a ratio $0<r<1$ we cluster the array into two clusters according to the value of unit distance. The first cluster comprises of the smallest elements of the array and its length is $K*r$, here we call it the *small-distance cluster*. The rest of the elements comprise the *large-distance cluster*.

If the distance of a unit belongs to the currently large-distance cluster, then we take all of its frames as part of the current extracted representative frames. If the distance of a unit belongs to the small-distance cluster, then we will delete all the frames except the first and the last frames from the unit. The first and the last frames are retained as part of the current extracted representative frames. After the deletion process, $K*r*(L-2)$ frames will be deleted.

Suppose the number of frames left is N''. If $N' \geq N''$, then our desired result is obtained and we can stop the algorithm. If it is not true, we can dynamically regroup all the retained frames as a new video and repeat the last procedure.

With the decrease in the number of frames for comparison, a unit will physically span across more frames in the original video. So it will adaptively represent a larger range of frame changes in the original video. The smaller the number we desire, the more times the algorithm would adaptively repeat the procedure. After each iterative process, there will be frames deleted from the sequence, so the overall number of frames left will decrease each time. Therefore, no matter how small a number may be required, the algorithm will adaptively and iteratively converge to the desired requirement.

Throughout the algorithm, shot boundaries do not need to be detected. Experiments show that it also will not be affected by gradual shot transitions(figure 7). The algorithm will automatically converge. So, an adaptive way of extracting is achieved. The whole process of our clustering algorithm is shown in figure 3.

3.2.2 Selection of Parameters

As the whole extraction process is basically unsupervised, the result will depend on the proper selection of the parameters L and r.

1) Selection of L

If L=2, the distance is in fact consecutive frame difference. Consecutive frame difference has been successfully applied for shot detection, but it is not suitable for finding representative frames. As illustrated in figure 1, assume on line GH are ten frames of a video. They are labeled 0 to 9. Their positions are shown as ticks on the line in the figure. Each step in 7 - 9 is larger than those in 0-7. Suppose we want to select 2 frames from G-H and use consecutive frame difference as measurement, we would delete all the frames in 0-7. However, the overall distance from 0-7 is actually even larger than that of 7-9, so we should extract at least one frame in 0-7. The failure of consecutive frame difference arises from the fact that it loses the cumulative change information in a video.

L	r	N_r	Representative Frames
3	0.3	16	0 67 68 78 109 110 114 118 169 170 258 259 265 300 306 307
5	0.3	17	0 67 68 69 70 113 114 115 116 168 265 299 300 301 305 306 307

Table 1.The Extracted Frames of "news", N'=17. N_r is the actually extracted number.

Generally speaking, if we use a large L, the algorithm will converge very fast and it will save a lot of computation time. In the beginning of the algorithm, a large L will not degrade result. However, if the required number is very small, the algorithm will iterate many times. With the iterations of the algorithm, the unit will in the end may physically span across many frames. Consequently, the smaller the L the better the result quality will be. Table 1 shows the frame numbers extracted from our test video "news". Its content is listed in table 2. Required representative number is $N'=330*0.05\approx17$. When L=3, the main information is indeed extracted by the algorithm but when L=5, frames of section 4 are all missed. In practice, if a video is very short, then we use L=3 in the algorithm. If the video is very long, then we use a variable L. In the beginning of algorithm, we let L=5 and when the extracted number drops to no more than 20% of original video, we then change L to 3.

2) Selection of r

If L=3 or 5, then 1 or 3 frames in each unit of the small-distance cluster will be deleted after the execution of one loop of the iterative algorithm. Accordingly, if before the iteration the retained number is N", then after the iteration, around:

$N''/2 * r * 1 = N'' * r * (1/2)$ for L =3, (6)

$N''/4 * r * 3 = N'' * r * (3/4)$ for L =5 (7)

number of the frames will be deleted.

In many cases, it is really not critical that the number of extracted representative frames is strictly equal to the required number. Assume that the maximum allowed error is 20%. Then we can calculate that the maximum allowed r is

$r = 0.2 / (1/2) = 0.4$ for L =3 (8)

$r = 0.2 / (3/4) \approx 0.3$ for L =5 (9)

Since the bigger the ratio r, the faster the algorithm converges, we try to use the largest r that we can possibly use in our algorithm. In practice, we select a r=0.3 in our work.

3.2.3 Experimental Results

We have run our algorithm on a lot of video data and the experimental results indicate that our algorithm is effective and robust. Here, we present two specific experimental results. The videos shown here are cut from long video sequences.

1) **Result on small-change sequences**

Table 2 shows the content of our test video "news". In section 1 and 4 there are only people speaking and the video content changes very little. This can also be seen in figure 5.a which shows the distances of first frame for all frames in the video. The result shows that though the two sections are much longer than other sections, the number of extracted representative frames from them are smaller than all other sections for all the fractions selected. Representative frames are extracted with respect to the changes of content selected, shown in figure 5.

Section	Range	Content	Section	Range	Content
1	0-67	Two anchor persons are speaking	4	170-258	An official is commenting
2	68-109	The first girl is receiving award	5	259-306	The second girl is receiving award
3	110-169	A man is receiving award	6	307-329	The third girl is receiving award

Table 2. The Content of "news".

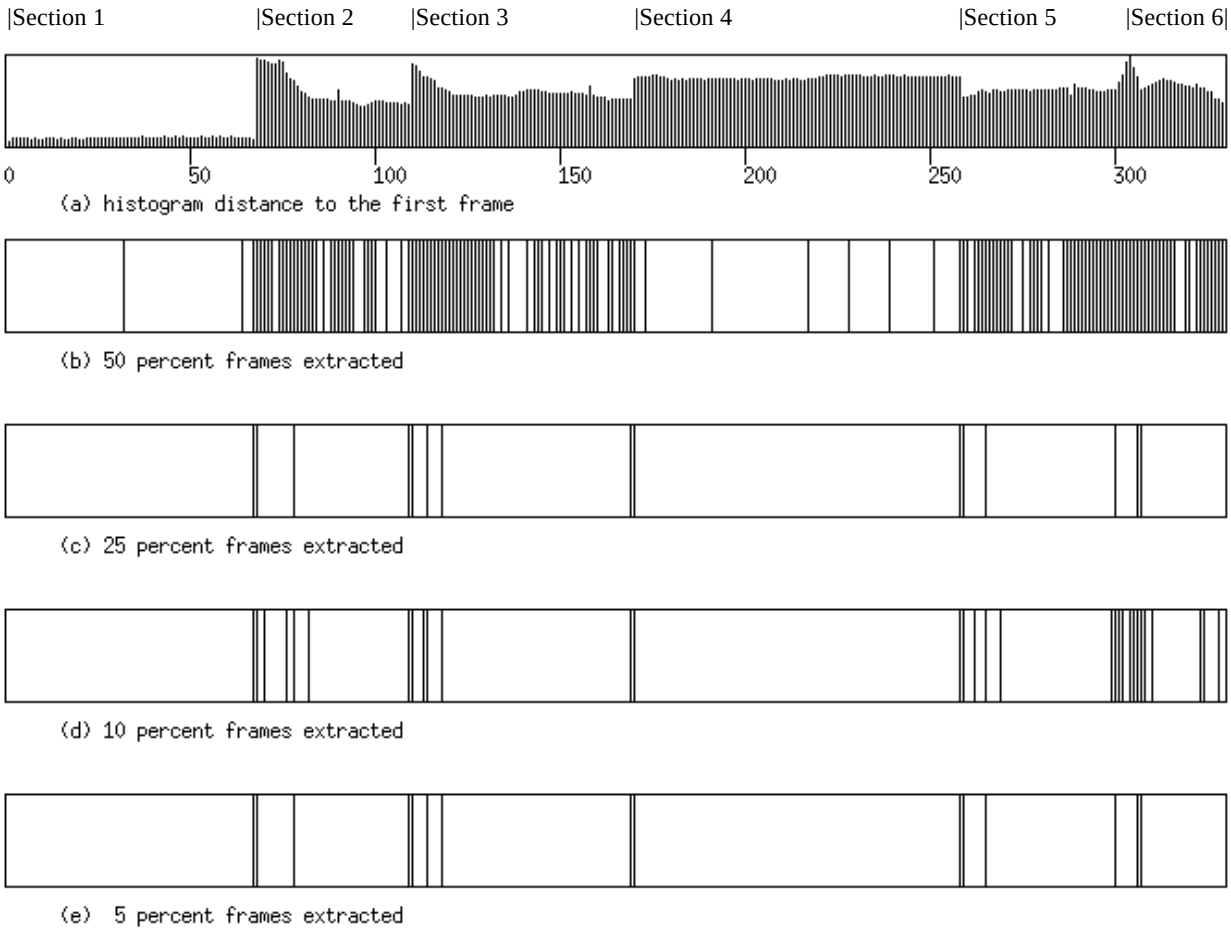


Figure 5. Experiment Result on "news".
The vertical lines in c, d, e and f represent the positions of extracted representative frames. L=2, r=0.3

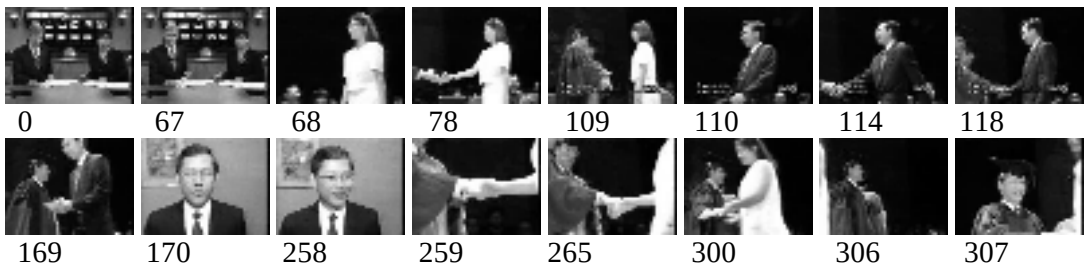


Figure 6. The Representative Frames of Figure 5.e in "news".

Section	Range	Content	Section	Range	Content
1	0-41	The first man' head is moving	4	148-205	The second man' head is moving
2	38-89	The first girl' head is moving	5	203-243	The third girl' head is moving
3	89-150	The second girl' head is moving			

Table 3. The Content of "mjackson".

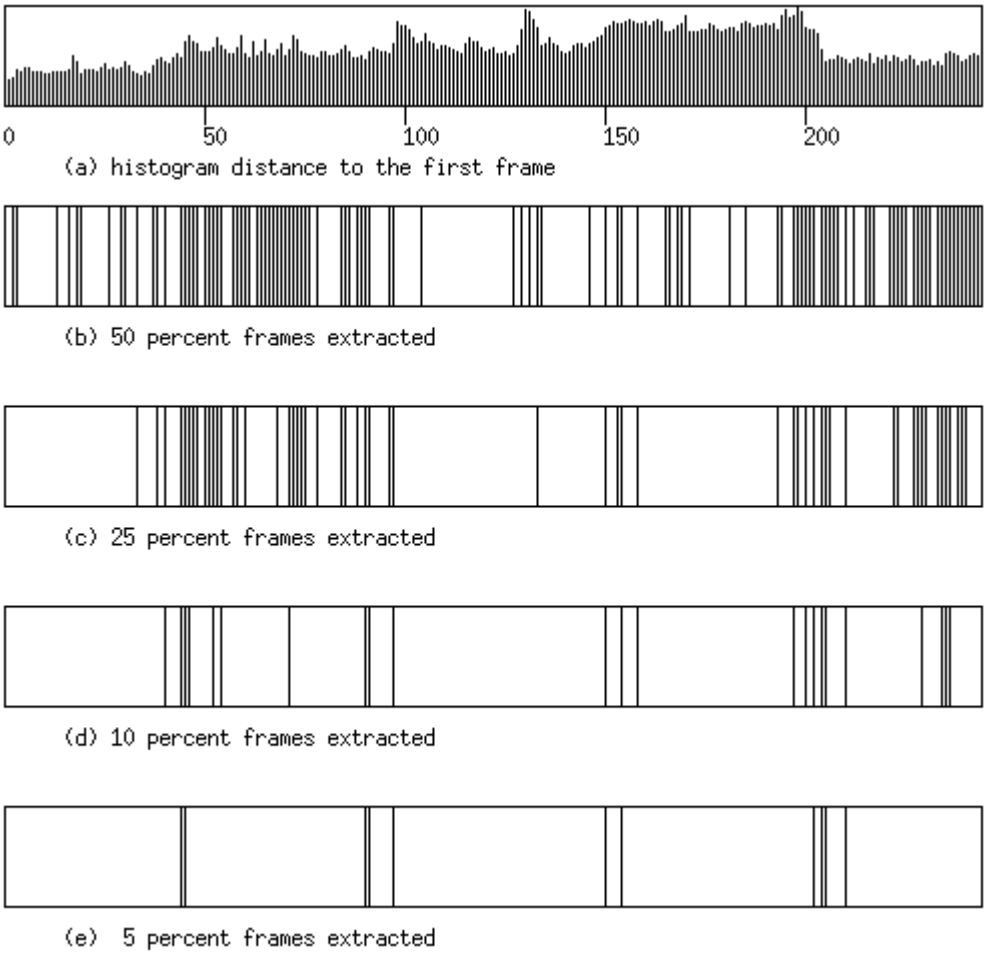


Figure 7. Experiment Result on "mjackson".
The vertical lines in c, d, e and f represent the positions of extracted representative frames. L=2, r=0.3

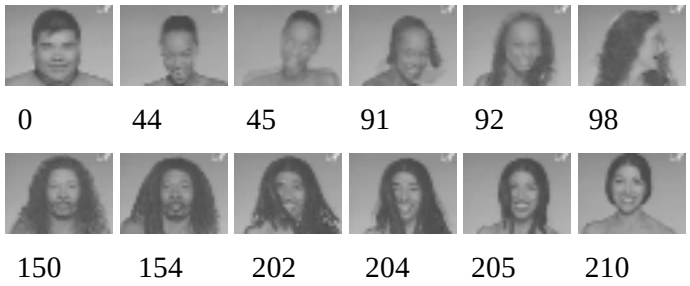


Figure 8. The Representative Frames in Figure 7.e of "mjackson".

2) **Result on gradual-change sequences**

The content of our test video “mjackson” is shown in table 3. The morphing technique has been used in the video sequence. The changes across the sections are gradual and it is very difficult to define precisely where the shot boundaries exist. So, we only give a rough border in the table. Without detection of shot boundaries, we successfully extract representative frames from the video according to content changes. When required number reaches 5% length of the original video, the main information of the video is still maintained very well.

4. APPLICATION OF REPRESENTATIVE FRAME EXTRACTION TECHNIQUE

To gather information from videos, we go back and forth between three stages: grazing, browsing and watching[11]. In the grazing stage, the user is passive, just waiting for interesting information to appear. In the browsing stage, a user interactively searches for information with no specific target in mind. In the watching stage, a user concentrates on understanding information. Since video data is usually very large, it puts a lot of demand on network resources for transmission. In order to save time, expense and bandwidth, before we download a video from the world wide web or just startup a remote on-line video and begin the understanding stage, it is worthwhile to spend some time getting a general idea about its content. After the extraction of representative frames, we obtain a subset which represents the important content of a video very well. It therefore provides a good basis for the development of video reviewing tools. Based on our content-based representative frame extraction technique, we have developed a *CBAC* video player system which has the functions of content based browsing and content-based video summary.

4.1 CONTENT-BASED VIDEO BROWSING

The functionality of content-based browsing is often compared to the function of a video cassette recorder(VCR). It is the next logical step beyond simple browsing using the fast-forward/rewind (FF/REW) buttons. After the development of shot detection techniques, many efforts have been made by researchers on how to provide an effective browsing tool. These efforts range from using a representative frame in a shot in Arman et al.[3], to the clustering of shots by Yeung et al.[12] and Zhong et al.[17], to the clustering of story units of Yeung et al.[13]. All of the development of such browsing techniques concentrates on providing a general structural description of a video.

In practice, a user mostly wants to use the VCR's fast forward and rewind functions. The user may just want to skip over some uninteresting sequence of frames when watching a video. Because within a shot there maybe many changes, skipping a shot will probably not meet the requirement of the user. The user in fact would like to skip to the next interesting part of a video.

Based on our *CBAC* technique, we make such a skipping function possible. As the representative frames with respect to a content have been computed and indexed, we can let the user select on-line his feature of interest and the percent of representative frames he would like to use for skipping. The user is thus provided with a very flexible tool

4.2 CONTENT-BASED VIDEO SUMMARY

Browsing tools provide interactive functions which help users to get hierarchical views of a video, from a global view to a detailed one. The interactive functionality of a browsing tool is very useful. However, the interaction process itself involves a lot of feedback between the user and the computer. There can be anywhere from 500 to 1000 shots per hour in a typical video program [1]. If one frame of each shot is selected from the video, it will compose a very large structure graph. So, to traverse the graph is still very time consuming for many end users. Therefore, many users would prefer a grazing view over the traversal of a browsing structure.

Zhang et al.[16] have proposed a way of finding representative frames within a shot. The extracted frames are output sequentially for video review. However, from the process of frames extraction we can see that these frames are of different content. From our experiments we have found that if a sequence of unrelated pictures is output sequentially

at the normal frame rate, the user will find it very difficult to grasp some information from it.

In our content-based video summary, we use the *representative sequences* which is composed of a representative frame plus its several successive frames to describe a video. The length of the following frames is called the *smoothing factor* and can be tuned by a user to obtain a smooth output.

If we use (1) to describe representative frames, then representative sequences can be described as :

$$\begin{aligned} V'' &= V'_1 \oplus V'_2 \oplus \cdots \oplus V'_{N'} \\ &= \{R_{p11}, R_{p12}, \ldots, R_{p1S}\} \\ &\quad \oplus \{R_{p21}, R_{p22}, \ldots, R_{p2S}\} \oplus \cdots \\ &\quad \oplus \{R_{pN'1}, R_{pN'2}, \ldots, R_{pN'S}\} \end{aligned} \quad (10)$$

where

- $S \geq 0$, is the smoothing factor
- $V'_i = \{ R_{pi}, R_{pi1}, \ldots, R_{piS} \}$, is the representative sequence of R_{pi} of feature P, $i \in [1,N']$
- $V'_1, V'_2, \ldots, V'_{N'}$ and $V'' \subseteq V$
- $\oplus$ is the temporal concatenation operation

So, given a video V, we can obtain the representative frames by using *CBAC* with an appropriately specified ratio. These representative frames can be augmented by "S" successive frames and this entire concatenated sequence constitutes the summary video of V. If S=0, then V"=V' and the result is the representative frame sequence. From our experiments, we find that to obtain a visually pleasing result, S=5 is the smallest smoothing factor required.

4.3 *CBAC* MPEG VIDEO PLAYER

The video browsing and summary technique has been incorporated in our *CBAC* system. The system works on the Sun Solaris system and has been written in C using Motif. The user interface of the system is shown in Figure 9.

The representative frames pre-computed at different percents and on different features are indexed for a given MPEG video. When a user opens a video file in the system, he has three choices: play, summary and browsing. If he only wants to play the video, he can set the "Ratio" to 100% and click the play button.

If the user wants to perform a content-based reviewing, then he has to select his content of interest by changing the "Feature" first, followed by selecting a ratio(<100%) of the representative frames he wants to use. In the case he wants to do content-based browsing, he may then just click on the FF/REW buttons to skip to the positions of his interest. Each time the user clicks the FF/REW button, the system will automatically jump to the next/last consecutive representative frame. If he wants a content-based video summary to be played, he has to select the smoothing factor as well. After the selection of a proper smoothing factor, he can press the play button and a representative sequence is displayed. The rightmost button on the top row displays the current frame number.

5. FUTURE WORK AND DISCUSSIONS

Given a certain video for description, different people may give different subjective interpretations. This is because there is no standard hierarchical information structure in a video. Automatic processing of video story plots is still beyond current computer vision techniques. Yet, it is useful to develop new techniques for content-based video processing with the objective of helping users manage video information as much as possible.

The extraction of representative frames is the key step in many areas of video processing. In this paper, we propose a general approach(CBAC) which uses an adaptive clustering technique for the content-based extraction of representative frames of a digital video without shot boundary detection. At first, we analyze all the frames of a video together for content changes. The changes of the content in iteratively increasing units are compared globally to determine which part of a video is important for description. The proposed approach has been applied to histogram analysis and shows promising results. We are also applying this technique into the analysis of motion, shape, texture and color content which will work together to provide more advanced functions.

Based on the content-based extraction of representative frames, we have developed a video

player which has the functions of content-based browsing and content-based video summary. Moreover, the computed extraction result also provides a sound basis for solving problems such as video sequence comparison and video retrieval. We are currently working on all these aspects. Also, the MPEG video data format packs much information in a compressed manner and operations on compressed digital video images may be 50 to 100 times faster than the corresponding algorithms operating on the uncompressed images[9]. Therefore, we are also working on applying the algorithm directly on the compressed video data.

ACKNOWLEDGMENT

This work is supported by the Real World Computing Partnership, which is funded by the Ministry of Trade and Industry of Japan.

REFERENCES

1. P. Aigrain, H. J. Zhang, D. Petkovic, "Content-based Representation and Retrieval of Visual Media: A State-Of-the-Art Review," *Multimeda Tools and Applications* 3(3): pp. 179-202,1996.
2. E. Ardizzone, M. L. Cascia, "Automatic Video Database Indexing and Retrieval," *Multimedia Tools and Applications*.
3. F. Arman, R. Depommier, A. Hsu, M. -Y. Chiu, "Content-based browsing of video sequences," *Proc. ACM Multimedia 94*, pp. 97-103,1994.
4. J. Bloedow, "Filmmaking Foundations," Focal Press, 1991.
5. J. S. Boreczky, L. A. Rowe, "Comparison of Video Shot Boundary Detection Techniques," *Storage and Retrieval for Image and Video Databases IV, Proc. of IS&T/SPIE 1996 Int'l Symp. on Elec. Imaging: Science and Technology*, pp. 170-179, 1996.
6. A. Hampapur, R. Jain, T. Weymouth, "Digital Video Indexing in Multimedia Systems," *Proc. Workshop on Indexing and Reuse in Multimedia Systems, American Association of Artificial Intelligence*, 1994.
7. http://www.real.com.
8. B. C. O'Connor, "Selecting Key Frames of Moving Image Documents: A Digital Environment for Analysis and Navigation," *Microcomputers for Information Management*, 8(2): pp. 119-133, 1991.
9. B. C. Smith, L. A. Rowe, "A New Family of Algorithms for Manipulating Compressed Images," *IEEE Computer Graphics and Applications*, 1993.
10. M. A. Smith, T. Kanade, "Video Skimming for Quick Browsing based on Audio and Image Characterization," *Technical Report No. CMU-CS-95-186*, School of Computer Science, Carnegie Mellon University, 1995.
11. Y. Taniguchi, A. Akutsu, Y. Tonomura, H. Hamada, "An Intuitive and Efficient Accesss Interface to Real-Time Incoming Video Based on Automatic Indexing," *Proc.ACM Multimedia 95*, pp. 25-33, 1995.
12. M. M. Yeung and B. Liu, "Efficient Matching and Clustering of Video Shots," *IEEE International Conference on Image Processing*, pp. 338-341, 1995.
13. M. M. Yeung, B. L. Yeo, B. Liu, "Extracting Story Units from Long Programs for Video Browsing and Navigation," *International Conference on Multimedia Computing and Systems*, 1996.
14. R. Zabih, J. Miller, K. Mail, "A Feature-Based Algorithm for Detecting and Classifying," *Proc. ACM Multimedia'95*, pp. 189-200, 1995.
15. H. J. Zhang, A. Kankanhalli, S. W. Smoliar, "Automatic partitioning of full-motion video," *Multimedia Systems*, 1(1): pp. 10-28,1993.
16. H. J. Zhang, C. Y. Low, S. W. Smoliar, "Video parsing and browsing using compressed data," *Multimedia Tools and Applications*, 1(1): pp.89-111, 1995.
17. D. Zhong, H. J. Zhang, S.-F. Chang, "Clustering Methods for Video Browsing and Annotation," *SPIE Conference on Storage and Retrieval for Image and Video*, 1996.
18. A. Nagasaka, Y. Tanaka, "Automatic Video Indexing and Full-Video Search for Object Appearances," *Proc. 2nd Working Conf. Visual Database Systems*, pp. 119-133 , 1991.

Figure 9. The CBAC MPEG Video Player.