Constructing Storyboards Based on Hierarchical Clustering Analysis

Satoshi Hasebe[a], Mustafa M. Sami[b], Shogo Muramatsu[b] and Hisakazu Kikuchi[b]

[a]Wireless & Visual Communications Co., Ltd., Niigata, Japan
[b]Department of Electrical and Electronic Engineering, Niigata University, Niigata, Japan

ABSTRACT

There are growing needs for quick preview of video contents for the purpose of improving accessibility of video archives as well as reducing network traffics. In this paper, a storyboard that contains a user-specified number of keyframes is produced from a given video sequence. It is based on hierarchical cluster analysis of feature vectors that are derived from wavelet coefficients of video frames. Consistent use of extracted feature vectors is the key to avoid a repetition of computationally-intensive parsing of the same video sequence. Experimental results suggest that a significant reduction in computational time is gained by this strategy.

Keywords: Video summarization, clustering, wavelets

1. INTRODUCTION

Video streaming has became one of major applications of broadband networks. Manifold networks, from error-free wired networks to error-subjected wireless networks, require highly scalable video codings. Motion-JPEG2000[1] is a wavelet-based intra-frame video coding with numerous features such as lossy-to-lossless seamless coding, progression order, ROI and so forth. Next generation scalable video codings feature motion compensated temporal filtering (MCTF)[2] to achieve higher coding efficiency as well as to provide frame rate (temporal) scalability, which are also wavelet-based codings.

There are growing needs for quick preview of video contents for the purpose of improving accessibility of video archives as well as reducing network traffics. Storyboards are one of helpful tools for video editors to plan the content composition and to browse video collections[3] as well as for shot-based video retrieval systems.[4] A storyboard consists of a series of video frames that are referred to as keyframes.

Keyframes are one of the most common representations for summarizing video shots. They are utilized for various video summarization applications[5] as well as flexible view of video contents[6,7] Keyframe generation is fundamental for video content analysis,[8] browsing[9] and retrieval techniques[10] that are based on frame features such as colors, shapes and textures.

Extracted initial keyframes are further refined to produce more efficient abstraction of a given video sequence. Zhong et al.[11] have proposed a generalized top-down hierarchical clustering process to construct hierarchical views of video shots. They have performed fuzzy partition clustering as well as standard k-means and hierarchical clustering on several types of feature vectors such as color features, temporal variances, and statistical motion features.

Clustering a large number of keyframes requires many computations. Lee et al.[12] uses RGB histograms as high dimensional feature vectors that are extracted from initial keyframes. Singular value decomposition[13] is applied to those feature vectors before k-means clustering is performed. This contributes to speeding-up the clustering. Drew et al.[14] defines *chromaticity signatures* of initial keyframes by a few bases to form low dimensional feature vectors. Those bases are prepared in advance according to the result of the singular value

Further author information: (Send correspondence to Satoshi Hasebe)
Satoshi Hasebe: hasebe@acm.org
Mustafa M. Sami: sami@telecom0.eng.niigata-u.ac.jp
Shogo Muramatsu: shogo@eng.niigata-u.ac.jp
Hisakazu Kikuchi: kikuchi@eng.niigata-u.ac.jp

Visual Communications and Image Processing 2005, edited by Shipeng Li, Fernando Pereira, Heung-Yeung Shum, Andrew G. Tescher, Proc. of SPIE Vol. 5960 (SPIE, Bellingham, WA, 2005) · 0277-786X/05/$15 · doi: 10.1117/12.631385

decomposition. Adjacent clusters are merged, before non adjacent clusters are merged. Finally, those frames of which feature vectors are closest to the cluster centers are selected as keyframes

There are several key issues for efficient and effective video summarization: shot boundary detection, keyframe extraction, keyframe clustering and dimensionality reduction. These are, in nature, independent to each other, and thus, different solutions are employed for individual problems to perform better. However, it sometimes requires unacceptably many computations, since every technique may analyze a given video sequence to extract different feature data. One solution to this problem is a mid-level representation[15] that offers a common platform for developing several techniques in video summarization.

In this paper, a storyboard that contains a user-specified number of keyframes is produced from a given video sequence. It is based on hierarchical cluster analysis[16] of feature vectors that are derived from wavelet coefficients of video frames. We have proposed several techniques such as shot boundary detection,[17] k-means clustering of keyframes[18] and video querying.[19] An underlying idea of our approach is to use the extracted feature vectors in common to avoid a repetition of computationally-intensive parsing of the same video sequence in different steps.

The rest of this paper is organized as follows. Section 2 describes several key techniques for video summarization including feature vectors, keyframes, and clustering. Section 3 describes experiments of storyboard production and performance evaluations in terms of accuracy and speed. Finally, Section 4 gives some concluding remarks on this work.

2. STORYBOARD CONSTRUCTION

This section presents a method to make a storyboard from a given video sequence. The method is characterized by the followings.

- Flexible view of a video sequence is realized with a storyboard that contains a user-specified number of keyframes.
- It offers an on-line reproduction capability for storyboards and this allows a user to find a better storyboard at his/her preference.
- Hierarchical cluster analysis is developed to group similar keyframes.
- Wavelet-based feature vectors are consistently utilized throughout a sequence of shot boundary detection, keyframe extraction and keyframe clustering.

For this purpose, a feature vector and the distance measure are defined. Then, each procedure of video summarization is described in detail.

2.1. Feature Vectors and Distance Measures

Every frame of a given video sequence is parsed to produce a feature vector. The feature vector is responsible to compute a certain distance between frames in shot boundary detection and in the following clustering analysis. For this purpose, similarity distance and the average feature vector are defined.

A feature vector

$$F = \{C, S\} \tag{1}$$

comprises the coarsest subband C and the significance map[17] S of finer subbands of the two dimensional wavelet transform of a frame picture. The coarsest subband consists of quantized wavelet coefficients. It is a coarse approximation of a frame. The significance map is a binary map: significant coefficients in finer subbands are encoded as unity and insignificant coefficients are encoded as zero. It implies the presence of sharp changes such as edges and textures.

The distance between two feature vectors, say F_m and F_n, is defined after some preliminary definitions. The L1 distance between two coarsest subbands, C_m and C_n, is described by

$$||C_m - C_n||_{L1} = \sum_i \sum_j |c_m(i,j) - c_n(i,j)| \tag{2}$$

where $c(i,j)$ denotes a quantized coefficient at (i,j) in the coarsest subband. The Hamming distance between two significance maps, S_m and S_n, is given by

$$||S_m - S_n||_H = \sum_i \sum_j \{s_m(i,j) \oplus s_n(i,j)\}, \tag{3}$$

where $s(i,j)$ denotes a binary at (i,j), and $\oplus$ represents exclusive OR. Finally, the distance between two feature vectors, $F_m$and F_n, is defined by a weighted sum of Eq.(2) and Eq.(3), as follows.

$$||F_m - F_n|| = w_0||C_m - C_n||_{L1} + w_1||S_m - S_n||_H, \tag{4}$$

where w_0 and w_1 are weights.

The cluster center is defined by the average of feature vectors in a cluster. With respect to a given set of multiple feature vectors. $F_1, F_2, \cdots, F_n$, a component of the average coarsest subband $\bar{C}$ over $C_1, C_2, \cdots, C_n$ is calculated by

$$\bar{c}(i,j) = \frac{1}{n}\sum_{k=1}^{n} c_k(i,j), \tag{5}$$

where $\bar{c}(i,j)$ denotes a coefficient at (i,j) in $\bar{C}$ and $c_k(i,j)$ denotes a coefficient at (i,j) in C_k. An average significance map $\bar{S}$ over $S_1, S_2, \cdots, S_n$ is calculated by

$$\bar{S} = T(H), \quad H = h(i,j), \quad h(i,j) = \sum_{k=1}^{n} s_k(i,j), \tag{6}$$

where $h(i,j)$ denotes the number of significant coefficients located at (i,j), and $s_k(i,j)$ is a binary at (i,j) in S_k. A mapping $T(\cdot)$ shows a thresholding-after-sorting operation as follows. After all elements $h(i,j)$ in H are sorted by their values in descending order, the largest N elements are quantized into unity and the others are quantized into zero.

As a result, a pair of the average coarsest subband and average significance map defines the average feature vector

$$\bar{F} = \{\bar{C}, \bar{S}\}. \tag{7}$$

2.2. Extraction of Initial Keyframes

Typical strategies for selecting initial keyframes are to select all frames in a video sequence, to select regularly subsampled frames and to select a set of initial keyframes according to the result of shot boundary detection. The first approach involves a large number of frames in clustering. Thus, the dimensionality of the feature vector should be small to complete clustering in practical time. Such a method is not qualified for on-line video analysis.[20] where processing speed is of great concern. The second approach simply reduces the number of frames. Redundant frames are removed by subsampling to some extent, since neighboring frames are very similar to each other. It can still leave redundant frames, and can falsely remove important frames, because it does not consider the frame contents at all. The last approach selects initial keyframes by considering the frame contents at the expense of computational cost. Our approach is based on the last one. It successfully reduces the computational cost by making wavelet domain-feature vectors in shot boundary detection, and they are kept to be used in clustering.

To analyze a given video sequence, the first step is shot boundary detection to find initial keyframes. We employ a two-step shot boundary detection algorithm,[17] which works in a wavelet transform domain. It captures gradual shot transitions as well as abrupt shot transitions. It computes a distance between video intervals to find isolated intervals. Then, it computes another distance between frames to find the exact location of a shot boundary. Both frame distance and interval distance are calculated by Eq.(4).

The two-step shot boundary detection algorithm have a single preference parameter to control the sensitivity of shot boundary detection. It is tuned to be a lower value to avoid detection misses. Although a lower value may cause quite a number of false positives, redundant frames are removed in the process of clustering. Hence,

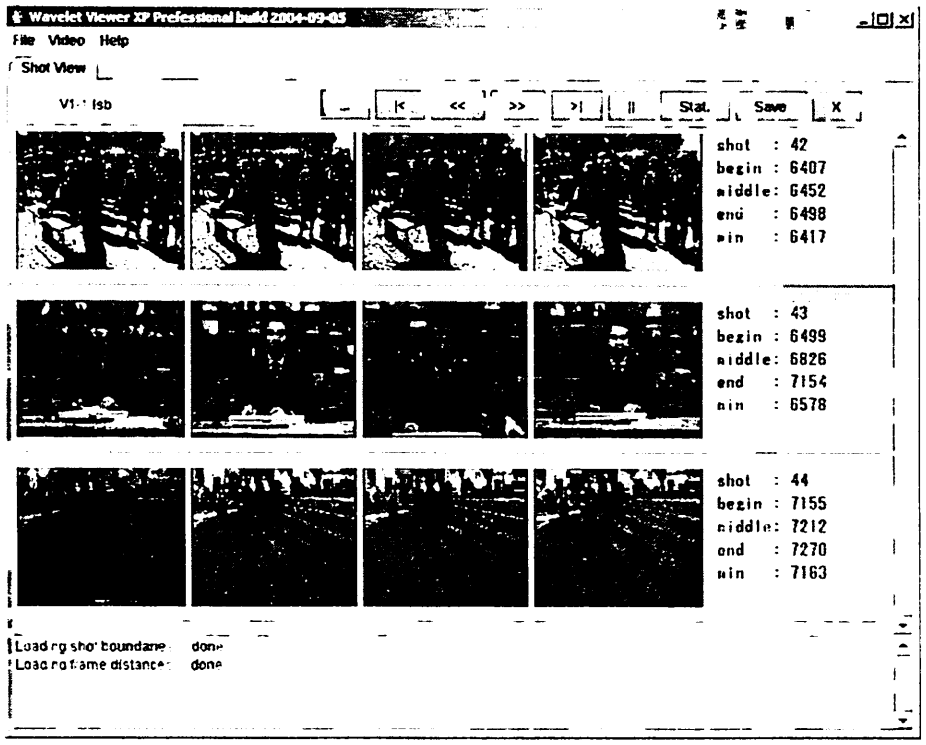

Figure 1. Keyframe candidates of shots.

it hardly influences the performance unless an extremely huge number of false positives. for example, almost the same as the total number of given video frames, have been produced.

After a given video sequence is divided into multiple shots, a single frame that well represents the shot content is selected for every shot. In psychological point of view, the first and the last frames in a shot convey vivid impressions. Unfortunately, a shot boundary is not always identified precisely. A frame adjacent to a shot boundary is likely to belong to a shot transition. Indeed, a gradual transition such as dissolve and fade consists of several frames. Such a frame that is involved with two different shot contents is not qualified for a representative frame.

We have tested the following four methods: to select the first and the last frame of a detected shot, respectively, to select a frame that is located at the midpoint, and to select frame that produces the minimum frame distance. As shown in Fig.1, the first two methods select several mixture frames as expected due to gradual shot transitions. The latter two methods have produced a fairly good result. The last method is slightly better and is robust against such a case that some shot boundaries are incorrectly detected, and the resulting false shot contains a transition within it. According to these observations, for every detected shot, we select a single frame that produces the minimum frame distance as an initial keyframe.

2.3. Hierarchical Clustering

A hierarchical cluster analysis is an alternative step to the previous k-means clustering. One of its advantages over the plain k-means clustering is to offer a fast clustering. It is performed as follows.

STEP1 For every pair of feature vectors, calculate the distance between them by Eq.(4). The results are stored in a distance matrix. Note that unlike k-means clustering, the number of initial clusters is equal to the total number of feature vectors, that is, every initial cluster contains a single feature vector.

STEP2 Search the distance matrix for the closest pair of clusters, and merge them into a new cluster. As a result, the total number of clusters decreases by one.

STEP3 Calculate the distances between the new cluster and the other clusters to update the distance matrix, which is detailed below.

STEP4 Repeat STEP2 and STEP3 until the total number of clusters reaches one, that is, all the feature vectors are contained in a single cluster.

There are several well-known methods to calculate the distance between clusters such as average linkage clustering, complete linkage clustering, single linkage clustering, and Ward's method.[21] In average linkage clustering, the distance between clusters is calculated based on averages such as a group average, centroid, and median. In complete linkage clustering, also known as furthest-neighbor method, the distance between clusters is defined as the maximum distance among all pairs of a feature vector in one cluster and that in another cluster. Similarly, in single linkage clustering, also known as nearest-neighbor method, the distance between clusters is defined as the minimum distance. Ward's method minimizes the total sum of squared distances between all feature vectors and the centroid for every cluster.

The Ward's method creates the tightest clusters, though it requires the Euclidean metric. The nearest neighbor method tends to produce relatively loose, chain-like clusters. The furthest neighbor method tends to form relatively tight clusters. Hence, it is favorable to video summarization and is applicable for the non-Euclidean metric Eq.(4).

As a result of cluster analysis, a binary tree referred to as dendrogram is obtained. It describes the structure of nested clusters. A cut point of the structuring tree is selected so that the desired number of sub trees are produced. For every sub tree, an average feature vector is calculated according to Eq.(6). Finally, for every cluster, a feature vector closest to the average is selected, and it shows up in the resulting storyboard.

3. EXPERIMENTAL RESULTS

We have implemented hierarchical and non hierarchical clustering algorithms for performance evaluations. In addition to a wavelet-based feature vector, we have chosen a *color histogram*[22] as a feature vector for comparison purpose. A histogram-based feature vector is often used as a color feature of a video frame.[11,12]

A color code histogram is generated as follows. After decoding a given video sequence, RGB color frame image is obtained. Each color band is assumed to have 8-bit color depth. For every color band, the most significant 2 bits are taken and they are combined to form a 6-bit color code. A 64-bin color code histogram is calculated for a given frame. A distance between two color code histograms is defined as the sum of absolute difference between corresponding bins. An average color code histogram is given by simply averaging corresponding bins.

Figure 2 shows a example of detected initial keyframes, and Fig.3 shows a screen shot of the resulting storyboard. 15 keyframes are specified by a user for summarizing the test video sequence.

It takes 8.8 seconds to perform hierarchical clustering analysis on 990-dimensional wavelet-based feature vectors of 522 keyframes. It takes 3.2 seconds to cut the structuring tree and to select a representative vector for every cluster. The total elapsed time to produce a storyboard amounts to 12 seconds.

A test video sequence includes various topics such as sports digest, on-the spot conference reports, and artist interviews. The total number of frames is 71,379, of which duration is about 48 minutes. It has 478 shots including 373 cuts, 90 dissolves, 7 wipes and the other special editing effects. 1020 shots are actually detected. These initial keyframes are refined with several clustering algorithms in previous sections.

Since shot boundary detection is not perfect, detected shots include some false positives that are undesirable for a storyboard. If just a single keyframe has been selected in a shot, and if it belongs to a stable non-transient interval, it is considered as a valid keyframe. If two or more keyframes are selected from a single shot, the first keyframe is considered as valid, and the other keyframes are considered as invalid. The total number of valid keyframes is, hence, equal to the number of true shots in a test video sequence. To evaluate the validity of selected keyframes, accuracy is defined by the percentage of valid keyframes among all of keyframes that have been actually selected.

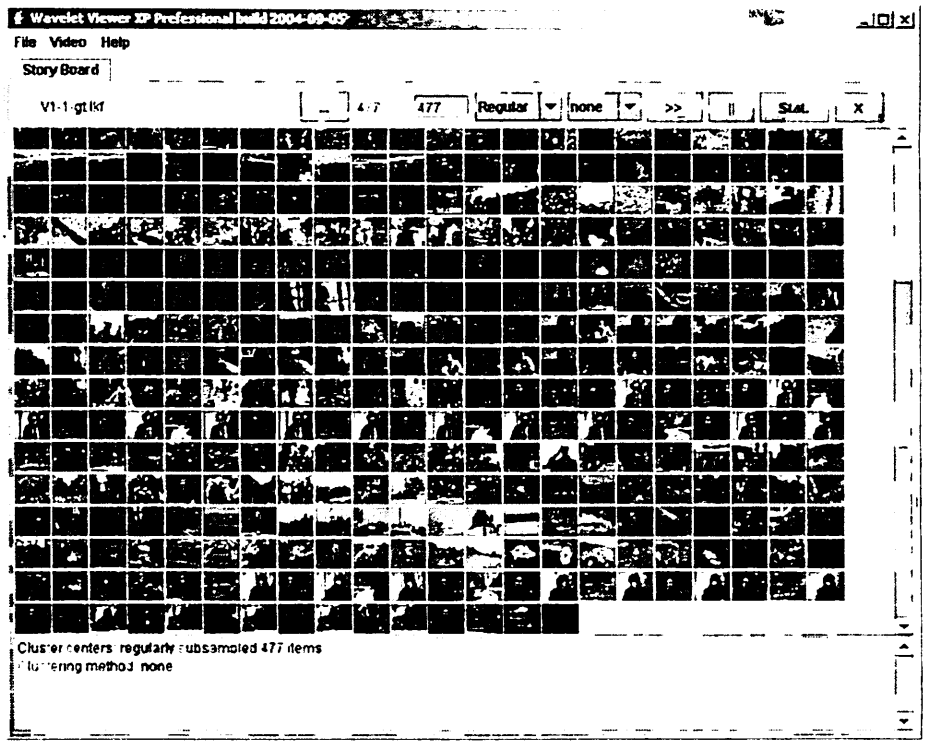

Figure 2. Detected initial keyframes.

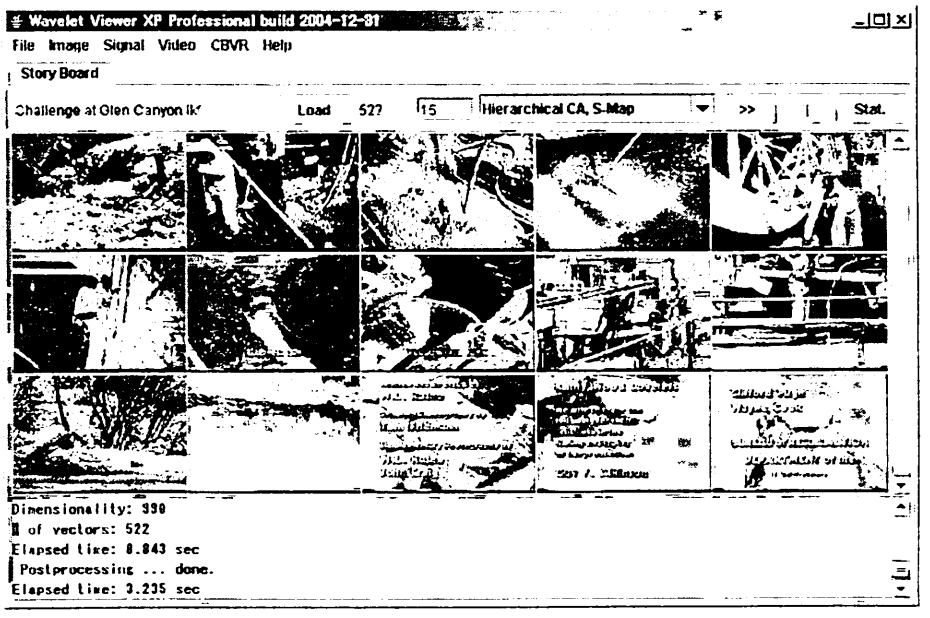

Figure 3. Screen shot of a storyboard.

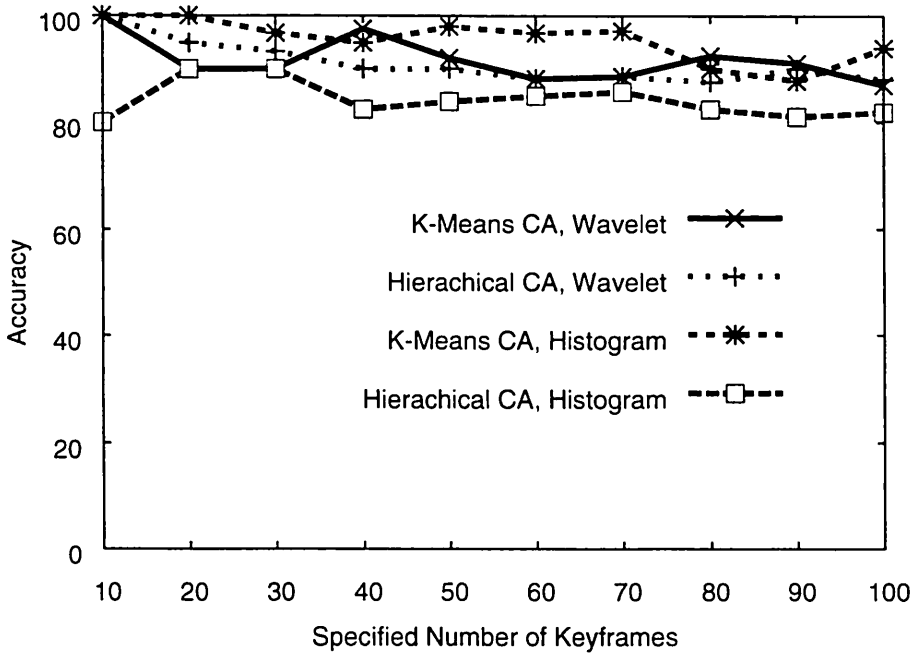

Figure 4. Accuracy versus the Number of Keyframes.

Table 1. Elapsed Time for Selecting 50 Keyframes

Method	Clustering	Pre/Post Processing
K-Means, Wavelet	310.2 sec	1.2 sec
Hierarchical, Wavelet	43.6 sec	4.6 sec
K-Means, Histogram	7.9 sec	41.2 sec
Hierarchical, Histogram	5.4 sec	83.6 sec

Figure 4 shows the accuracy versus the number of keyframes. Hierarchical clustering with histogram-based feature vectors produces unsatisfactory performance. On the other hand, hierarchical clustering with wavelet-based feature vectors is comparable to k-means clustering methods.

Figure 5 shows the total elapsed time in making a storyboard versus the number of keyframes. As the number of keyframes increases, the elapsed time in k-means clustering becomes longer while that in hierarchical clustering is almost identical. The k-means clustering with 60 keyframes converges after 4 iterations, while 8 iterations are required for 50 keyframes. This is why the elapsed time for 60 keyframes is shorter than that for 50 keyframes. Generally, in a k-means clustering algorithm, both the accuracy of clustering and the number of iterations depend on feature vectors and a choice of initial cluster centers.

Table.1, shows the elapsed time for clustering and the other processes in selecting 50 keyframes. The dimensionality of wavelet-based feature vector is 15 times as high as that of color code histogram. Thus, it is natural that wavelet-based methods are slower in clustering than histogram-based methods. In spite of this fact, the proposed hierarchical clustering methods is fastest in total elapsed time among all. Thanks to the consistent use of extracted feature vectors, it avoids computational intensive parsing of a video frame. Such a consistent use of color code histogram does not works well, since accurate shot boundary detection is not always achieved merely with the color code histogram.[17]

Once hierarchical clustering has been completed to generate the dendrogram, what is needed in recalculation with different destination number of keyframes is just to select representative keyframes for each cluster. It will be completed in a few seconds. This is very preferable for on-line reproduction of a storyboard.

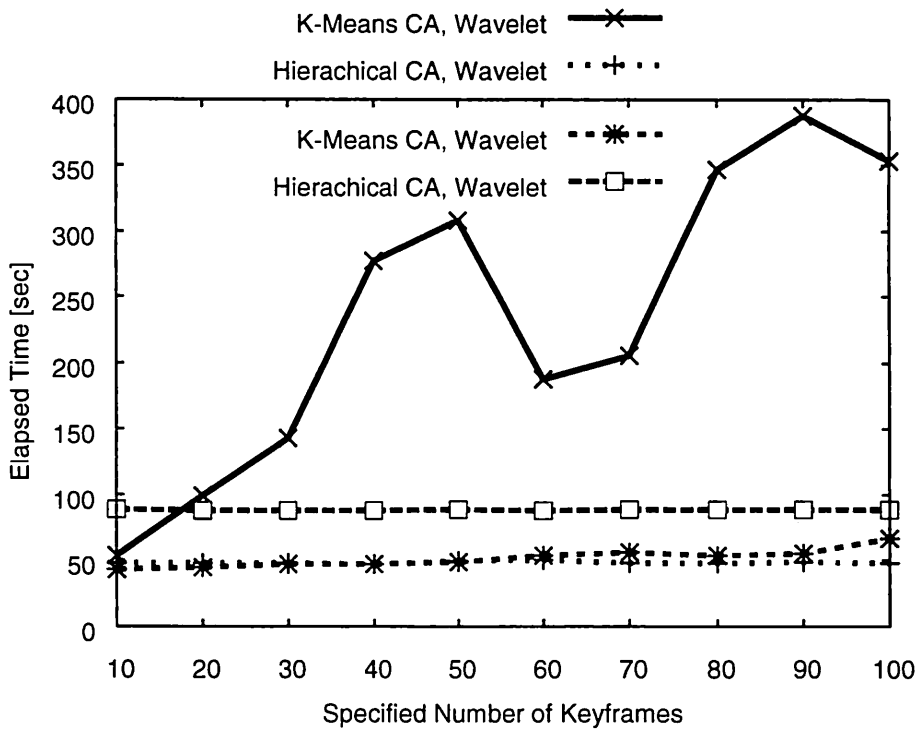

Figure 5. Elapsed Time versus the Number of Keyframes.

4. CONCLUDING REMARKS

We have proposed a keyframe refinement method based on hierarchical clustering analysis. The experimental results suggest that the hierarchical clustering with wavelet-based feature vectors offers satisfactory results. A significant reduction in computational time and memory requirements is gained by the consistent use of extracted feature vectors.

REFERENCES

1. W. Yu. R. Qiu and J. Fritts, "Advantages of Motion-JPEG 2000 in Video Processing," *SPIE Photonics West, Electronic Imaging 2002*. San Jose, CA, Jan. 2002.
2. Y. Andreopoulos, A. Munteanu, M. van der Schaar, J. Cornelis and P. Schelkens. "Comparison between "t+2D" and "2D+t" architectures with advanced motion compensated temporal filtering," ISO/IEC JTC1/SC29/WG11, m11045, MPEG 69th meeting. Redmond, US. July 2004.
3. P. J. Macer, P. J. Thomas, N. Chalabi and J. F. Meech, "Finding the Cut of the Wrong Trousers: Fast Video Search using Automatic Storyboard Generation." *Proc. Conference on Human Factors in Computing Systems*, pp.303–304, Vancouver, 1996.
4. M. Christel and N. Moraveji. "Finding the Right Shots: Assessing Usability and Performance of a Digital Video Library Interface." *Proc. 12th annual ACM international conference on Multimedia*, pp.732–739, New York, 2004.
5. M. Christel, A. Hauptmann, A. Warmarck and S. Crosby. "Adjustable Filmstripes and Skims as Abstractions for a Digital Video Library," *Proc. IEEE Advances in Digital Libraries Conference*. Baltimore. 1999.
6. S. Uchihashi. J. Foote, A. Girgensohn, and J.Boreczky. "Video Manga: Generating Semantically Meaningful Video Summaries." *Proc. Seventh ACM International Conference on Multimedia*. pp.383–392, Orlando. 1999.

7. J. Boreczky, A. Girgensohn, G. Golovchinsky and S. Uchihashi, "An Interactive Comic Book Presentation for Exploring Video," *Proc. SIGCHI Conference on Human Factors in Computing Systems*, pp.185–192, The Hague, 2000.
8. L. Chaisorn, T. S. Chua and C. H. Lee, "Segmenting Stories in News Video," *Handbook of Video Databases Design and Applications*, pp.1133–1148, CRC Press, Florida, 2004.
9. F. C. Li, A. Gupta, E. Sanocki, L.-W. He and Y. Rui, "Browsing Digital Video," *Proc. SIGCHI conference on Human factors in computing systems*, pp.169–176, The Hague, 2000.
10. X. Wen, T. D. Huffmire, H. H. Hu, and A. Finkelstein, "Wavelet-Based Video Indexing and Querying," *Multimedia Systems*, Vol.7, Issue.5, pp.350–358, Springer-Verlag, Heidelberg, 1999.
11. D. Zhong, H. Zhang and S. F. Chang, "Clustering Methods for Video Browsing and Annotation," *Proc. IS&T SPIE Symposium on Storage and Retrieval for Image and Video Database*, San Jose, 1996.
12. S. Lee and M. H. Hayes, "A Fast Clustering Algorithm for Video Abstraction," *Proc. IEEE ICIP*, Barcelona, 2003.
13. S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, San Diego, 1999.
14. M. S. Drew and J. Au, "Video Keyframe Production by Efficient Clustering of Compressed Chromaticity Signatures," *Proc. Eighth ACM International Conference on Multimedia*, pp.365-367, Marina del Rey, 2000.
15. L.-Y. Duan, M. Xu, T.-S. Chua, Q. Tian and C.-S. Xu, "A Mid-level Representation Framework for Semantic Sports Video Analysis," *Proc. the eleventh ACM international conference on Multimedia*, pp.33–44, Berkeley, 2003.
16. A. K. Jain, M. N. Murty and P. J. Flynn, "Data Clustering: A Review," *ACM Computing Surveys*, Vol.31, No.3, September, 1999.
17. S. Hasebe, S. Muramatsu, S. Sasaki, J. Zhou and H. Kikuchi, "Two-Step Algorithm for Detecting Video Shot Boundaries in a Wavelet Transform Domain", *Proc. Third International Symposium on Image and Signal Processing and Analysis*, pp.245–250, Rome, 2003.
18. S. Hasebe, M. Nagumo, S. Muramatsu and H. Kikuchi, "Video Key Frame Selection by Clustering Wavelet Coefficients," *Proc. EUSIPCO*, No.1679, pp.2303–2306, Vienna, Austria, 2004.
19. S. Hasebe, S. Muramatsu, S. Sasaki, H. Kikuchi, "Video Querying Based on Three-Dimensional Wavelet Transforms," *Proc. ITC/CSCC*, pp.1196–1199, Tokushima, 2001.
20. W. Zhou and C. C. J. Kuo, *Intelligent Systems for Video Analysis and Access over the Internet*, Prentice Hall, New Jersey, 2002.
21. G. N. Lance and W. T. Williams, "A General Theory of Classificatory sorting strategies," *Computer Journal*, Vol.9, Issue 4, pp.373–380, 1967.
22. H. J. Zhang, A. Kankanhalli and S. W. Smoliar, "Automatic Partitioning of Full-Motion Video," *ACM Multimedia Systems Journal*, pp.10–28, 1993.