Hierarchical Shot Clustering for Video Summarization

YoungSik Choi, Sun Jeong Kim, and Sangyoun Lee

Multimedia Technology Research Laboratory, Korea Telecom,
Seocho-Gu Woomyeon-dong 17
Seoul, Korea,
{choimail, sunjkim, leesy}@kt.co.kr

Abstract. Digital video is rapidly becoming a communication medium for education, entertainment, and a variety of multimedia applications. With the size of the video collections growing to thousnads of hours, efficient searching, browsing, and managing video information have become of increasing importance. In this paper, we propose a novel hierarchical shot clustering method for video summarization which can efficiently generate a set of representative shots and provide a quick and efficient access to a large volume of video content. The proposed method is based on the compatibility measure that can represent correlations among shots in a video sequence. Experimental results on real life video sequences show that the resulting summary can retain the essential content of the original video.

1 Introduction

With the recent advances in compression and communication technologies, vast amounts of video information are created, stored, and transmitted over networks for education, entertainment, and a host of multimedia application. Therefore, efficient searching, browsing, and managing video information have become of increasing importance. The MPEG group has recently begun a new standardization phase for efficient searching and managing multimedia content (MPEG-7). The MPEG-7 will specify the ways to represent multimedia information by means of descriptors and description schemes. The question of how to obtain these descriptors automatically is becoming a highly important research topic. Particularly, automatic video summarization is gaining the attention as a way to condense a large volume of video into smaller and comprehensible units, and allows quick and easy access to video content.

There are a few approaches to video summarization: (1) selecting and concatenating the "most representative" images or shots [3], [6]. (2) creating a "skim" video which represents a short synopsis [7]. In this paper, we address the problems with selecting the "representative" shots and propose a novel hierarchical shot clustering for video summarization.

Shot clustering has been frequently used for video summarization and segmentation. In

P.M.A. Sloot et al. (Eds.): ICCS 2002, LNCS 2331, pp. 1100–1107, 2002.

this approach, a video sequence is first segmented into shots and then shot clustering is applied to select the representative shots [3], [6] or to group the shots into scenes (story units) [4], [5]. Agglomerative hierarchical clustering with time constraint has been used for shot clustering [1], [2]. This approach can produce the tree-structured representation that is useful for video summarization. However, it is expensive, requiring insertions and deletions of the cluster dissimilarity matrix, and requiring a search for each closest cluster pairing. Window-based shot grouping has also been used [4], [5]. In this paradigm, incoming shots are compared with the shots in a given window in order to check if incoming shots may be included in the current segment. This method is computationally less expensive to group shots into scenes than agglomerative clustering. It is, however, difficult to have a compact hierarchical representation for video summarization.

To overcome the limitations in both approaches, we propose a novel shot clustering algorithm that can efficiently produce the representative shots and extract the hierarchical structure from a video sequence. The proposed clustering algorithm is based on the assumptions that the shots from a scene are more likely to be compatible with each other than those from other scenes, and that the shot highly compatible with other shots is more likely to be the representative shot of a scene. We define the compatibility measure to be the average value of the degrees to which a shot is similar to its neighboring shots within a given window. Using the compatibility measure, we develop an efficient clustering algorithm for video summarization.

2 Hierarchical Shot Clustering

2.1 Compatibility Measure

We define the compatibility measure as follows. Let $S = \{s_1, s_2, \ldots, s_k, \ldots, s_N\}$ denote a video sequence, where s_i is i-th shot and N is the total number of shots. The compatibility measure $C(s_i)$ of s_i within a given window is defined as:

$$C(s_i) = (1/\, N(i))\ \textstyle\sum_{j\in N(i)} \mu_{ji}, \qquad \textbf{(1)}$$

where $N(i)$ is the set of the neighbors of s_i and μ_{ji} is a fuzzy membership function which determines the value of the degree to which s_i is similar to s_j. $N(i)$ may be $\{s_{i-2}, s_{i-1}, s_{i+1}, s_{i+2}\}$ if the window size is 4. The membership function can be defined as a monotonically decreasing function of the dissimilarity between s_i and s_j. In this Letter, we propose to use the following bell-shaped membership function:

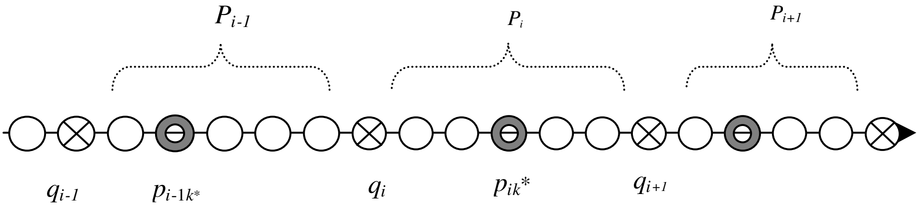
$$\mu_{ji} = \exp(-D(s_i, s_j)/\beta_j). \qquad \textbf{(2)}$$

Note that in general, μ_{ij} is not equal to μ_{ji}. The $D(s_i, s_j)$ denote the dissimilarity between s_i and s_j. The β_j is a scaling factor and is determined for each s_j to consider the local context as follows.

$$\beta_j = (1/|N(j)|)\Sigma_{i\in N(j)}D(s_i, s_j), \quad (3)$$

where $|N(j)|$ is the number of neighbors of s_j. The dissimilarity $D(s_i, s_j)$ between s_i and s_j may be defined as a function of shot feature vectors. In this Letter, the shot feature vector is computed as the average of all frame feature vectors within a shot.

Fig. 1. Illustration of grouping and handling local minima



2.2 Clustering Algorithm:

The proposed compatibility measure can represent the correlations among shots in a video sequence. If the value of $C(s_i)$ is higher than the values of its neighbors, s_i is more compatible with its neighbors and is, therefore, more likely to be the representative of its neighbors. If the value of $C(s_i)$ is lower than its neighbors, s_i is less compatible with its neighbors and, therefore a scene boundary more likely exists around s_i. Taking these into accounts, we propose the following hierarchical clustering method.

(Step 1) *Initialization*: Let $S = \{s_1, s_2, \ldots, s_k, \ldots, s_N\}$ be a video sequence and $\{\boldsymbol{h}(s_1), \boldsymbol{h}(s_2), \ldots, \boldsymbol{h}(s_k), \ldots, \boldsymbol{h}(s_N)\}$ be the corresponding shot feature vectors, where $\boldsymbol{h}(s_k)$ is the feature vector of shot s_k. We initialize S as the set of initial clusters, P.

(Step 2) *Grouping and selecting a key shot*: For each cluster in P, we obtain the compatibility using equations (1), (2), and (3). We find the clusters with local minimum compatibility values. Let $\{q_1, q_2, \ldots, q_n\}$ denote the set of these clusters, where n is the total number of local minima (See Figure 1).
We group the clusters between clusters q_i and q_{i+1} into a new cluster P_i. Let $P_i = \{p_{i1}, \ldots, p_{ik}^*, \ldots, p_{in}\}$, where *in* is the total number of clusters between q_i and q_{i+1}, and p_{ik}^* is the cluster with the local maximum compatibility value between q_i and q_{i+1}. Note that only one p_{ik}^* exists between q_{i-1} and q_i. We select p_{ik}^* as the representative cluster of P_i.

If this is the first iteration, we select p_{ik}* as the key shot of P_i. Otherwise, we select p_{ik}*'s key shot as the key shot of P_i. Note that the representative clusters and the key shots are the same in the first iteration.

(Step 3) *Handling Local Minima*: We can consider $\{q_1, q_2, \ldots, q_n\}$ as outliers or boundary clusters. Therefore, we first check whether q_i is an outlier or a boundary cluster according to $C(q_i)$. If q_i is an outlier, q_i becomes a new cluster between $P_{i\text{-}1}$ and P_i. Otherwise, we merge q_i into the closer cluster $P_{i\text{-}1}$ or P_i with respect to the dissimilarity function defined in equation (4). The following describes this process.
For each local minimum cluster q_i starting from q_1, do the following.
(Step 3-a) If $C(q_i) < T_C$, then make q_i as a new cluster between $P_{i\text{-}1}$ and P_i. T_C is the threshold value.
(Step 3-b) If $C(q_i) \geq T_C$, do the following.
If $\mathrm{A}(q_i, P_{i\text{-}1}) < \mathrm{A}(q_i, P_i)$, then add q_i into cluster $P_{i\text{-}1}$. Otherwise, add q_i into cluster P_i. $\mathrm{A}(q_i, P_i)$ is the dissimilarity between clusters q_i and P_i, and defined as

$$\mathrm{A}(q_i, P_i) = \min D(q_i, p_{ij}), \text{ where } p_{ij} \in P_i. \quad \mathbf{(4)}$$

(Step 4) *Time constraint and terminating condition*: For each new cluster P_i, we check the time constraint with threshold value T_T. If the duration of cluster exceeds T_T, then we ungroup the cluster and make the clusters in P_i as new clusters between $P_{i\text{-}1}$ and P_{i+1}. Note that this ungrouping is similar to clustering with time-constrained distance as in [1]. We terminate the clustering process if there is no change in clusters P_i.

(Step 5) *Update cluster feature vector*: We update the feature vectors of new clusters P_i such that the feature vectors of more compatible clusters may gain more weights than those of less compatible clusters. That is, we update the feature vector $\boldsymbol{h}(P_i)$ for each cluster P_i as the weighted average of feature vectors $\boldsymbol{h}(p_{ij})$ of clusters p_{ij} in P_i.

$$\boldsymbol{h}(P_i) = \Sigma C(p_{ij})\boldsymbol{h}(p_{ij})/\Sigma C(p_{ij}),$$

where $p_{ij} \in P_i$. Set $P = \{P_1, \ldots, P_m\}$ and m is the number of new clusters P_i. Go to Step 2.

3 Video Summarization

The proposed clustering algorithm results in a hierarchical structure of a video sequence where each node corresponds to P_i. Each node has the representative cluster p_{ik}* and the key shot (See Step 2 in 2.2). The set of clusters P_i in the highest level can be considered as a partition of a video sequence (a video segmentation) and the set of the representative clusters p_{ik}* and key shots as a abridged version (a video summarization). Taking these into accounts, we present the following scheme for video summarization.

Suppose that our clustering method produces M clusters in the top level and we select the set of key shots in each cluster as a summary. Then, the length T_C of the summary becomes

$$T_C = \Sigma_{i=1..\ M}\ T(S_i),$$

where $T(S_i)$ is the length of key shot S_i from cluster P_i. Conversely, if a user requests a summary of length T_{req}, we need to determine the number of clusters in the top level. Assuming that $T(S_i)$ be the average shot length T_{avg} in a video sequence, the number of segments M becomes

$$M = T_{req}/T_{avg}.$$

Now, we need to adjust the threshold value T_T to produce M clusters in the top level. In the proposed clustering algorithm, the number of clusters M has an inverse relation to the threshold value T_T in Step 4 in Section 3. Therefore, we propose to set T_T as

$$M \qquad \mathbf{5}$$

where T_{org} is the length of the original video and k is a constant greater than 1.

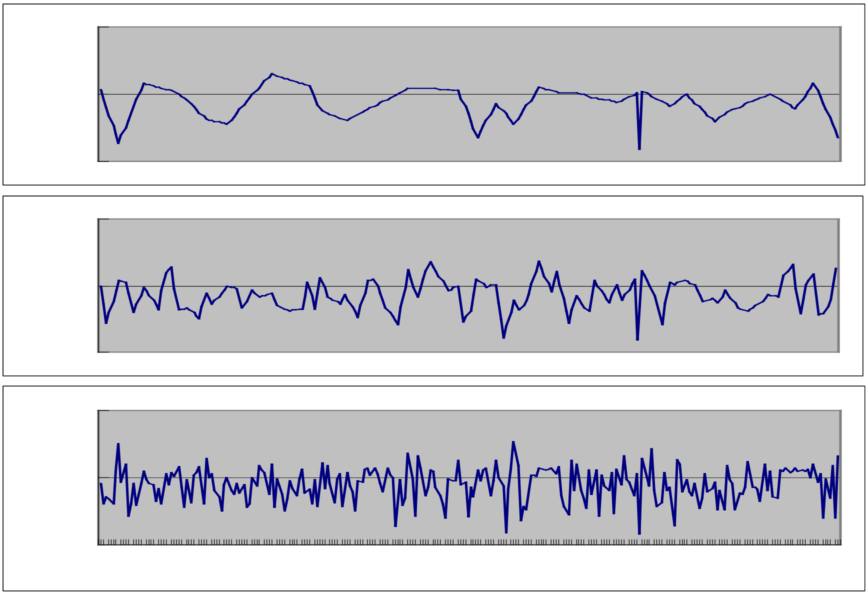
Using equation (5), we can generate a summary as a user requests in terms of T_{req}, the length of the summary. Let $M\,'$ be the actual number of clusters in the top level produced by using the threshold value in (5). Then, we can simply generate a summary by selecting the set of key shots from $M\,'$ clusters.

4 Experimental Results

To test the proposed clustering and summarization method, we used a 55 minutes and a 45 minutes TV dramas. We segmented the 55 minutes drama into 399 shots and the 45 minutes drama into 291 shots using traditional histogram difference. We used mean color histogram for a shot feature vector as in [5].

We set the threshold values T_C and the window size for the compatibility computation as 0.1 and 4, respectively. With these values and T_T of 250 seconds, our clustering converged within 3-4 iterations. Figure 2 shows the compatibilities of the TV drama with window size 4. In this Figure, the peaks correspond to the representative shots and the valleys correspond to the possible scene boundary shots.

Fig. 2. Compatibilities versus shot numbers: (a), (b), and (c) compatibilities after 2, 1, and 0 iterations, respectively.



In order to test the effectiveness of equation (5), we varied T_{req} and compared the desired number of clusters with the actually generated number of clusters. Table 1 shows the results obtained from the two TV dramas with k = 2 in equation (5). The results show that the number of generated clusters is close to the desired number of clusters.

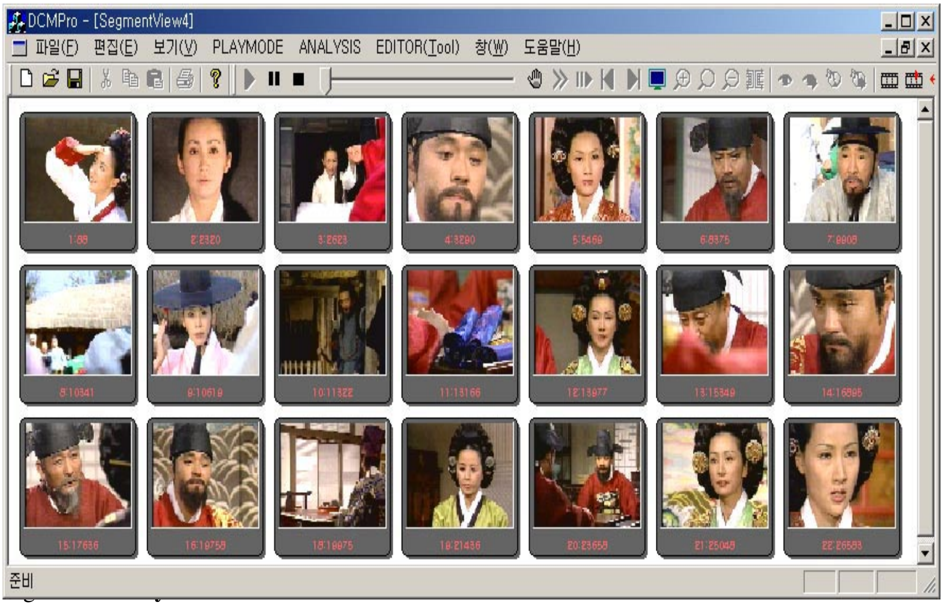
Table 1. Comparison of the desired and generated number of clusters with respect to T_{req}

T_{req} (Request length in minute)	TV Drama I (399 shots: 55 minutes)			TV Drama II (291 shots: 45 minutes)		
	C (Desired number of clusters)	T_T (Threshold value in second)	C' (Generated number of clusters)	C (Desired number of clusters)	T_T (Threshold value in second)	C' (Generated number of clusters)
2	14	480	14	12	450	9
3	21	320	21	19	284	22
4	28	240	31	25	216	33
5	35	192	31	32	169	36
10	71	95	72	64	84	71

We defined the number of key shots coming from the different scenes as the performance measure of the proposed summarization method. That is, the summary is better if it represents more scenes in the original video. For this experiment, we obtained the ground truth scene boundaries by manual segmentation. Table 2 shows the results. In Figure 3, we show the summary result of TV Drama I. The images shown in this Figure are the first frames of the selected key shots.

Table 2. Performance measure

T_{req} (Request length in minute)	TV Drama I (399 shots: 30 scenes)		TV Drama II (291 shots: 27 scenes)	
	C' (Generated number of clusters)	Number of scenes that the key shot represents	C' (Generated number of clusters)	Number of scenes that the key shot represents
2	14	12	9	9
3	21	15	22	17
5	31	23	33	24



5 Conclusions

In this paper, we presented a hierarchical clustering method based on the compatibility measure that can represent shot correlations. The proposed method can efficiently generate a set of representative shots and also extract the hierarchical structure of a video sequence. Experimental results show that our proposed summarization abridged the original video where compaction is up to 25:1 and still kept most of important scenes. This result is accredited to the clustering capability of extracting the hierarchical structure of a video sequence.

References

1. M Yeung and Boon-Lock Yeo, and Bede Liu “Extracting Story Units from Long Programs for Video Browsing and Navigation”, Proceedings of IEEE International Conference on Multimedia Computing and Systems1996, pp. 296-305.
2. E. Venequ and et al, “From Video Shot Clustering to Sequence Segmentation”, Proceedings of IEEE International Conference on Pattern Recognition 2000, pp. 254-257.
3. Shingo Uchihashi and, et al, “Video Magna: Generating Semantically Meaningful Video Summaries”, Proceedings of ACM International Conference on Multimedia 1999, pp. 383-391
4. A. Hanjalic and et al, “Automated High-Level Movie Segmentation for Advanced video-Retrival Systems”, IEEE Transactions On Circuits and Systems for Video Technology, Vol. 9, No. 4, June 1999, pp. 580-588.
5. Ong Lin and Hong-Jiang Zhang, “Automatic Video Scene Extraction by Shot Grouping”, Proceedings of IEEE International Conference on Pattern Recognition 2000, pp. 39-42.
6. Nikolaos D. Doulamis and et al, “Efficient Summarization of Stereoscopic Video Sequences”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 10, No. 4, June 2000, pp. 501-517.
7. Michael A. Smith and Takeo Kanade, “Video Skimming and Characterization through the Combination of Image and Language Understanding Techniques”, Proceeding of IEEE, pp.775-781. 1997