ACCV2002: The 5th Asian Conference on Computer Vision, 23-25 January 2002, Melbourne, Australia.

A Scene-Level Analysis for Video Abstraction

Hang-Bong Kang

Dept. of Computer Engineering, The Catholic University of Korea
#43-1 Yokkok 2-dong, Wonmi-Gu, Puchon City, Kyongki-Do, Korea
E-mail: hbkang@www.cuk.ac.kr

ABSTRACT

In this paper, we propose a novel scene-level analysis method for video abstraction. Video abstraction refers to the compact representation of a long original video and is useful for a user to decide whether the video is worth to viewing or not. To effectively construct video abstraction, it is necessary to understand video data semantically. One possible approach to understand semantics in video data is to compute the contextual information within a scene and inter-scenes. We divide the context in the scene into local context and global context. We define the local scene context as the context within the scene such as the shots' coherences and interactions in the scene. We also define the global scene context as the context of inter-scenes such as the similarity amongst scenes and relative importance of the scene in composing video data. We compute contextual information at the scene-level and detect dominating scenes for video abstraction. Experimental results are presented.

Keywords: video abstraction, scene-level analysis, contextual information, semantics

1. Introduction

Video abstraction is the compact representation of a long video data and is useful to a viewer to decide whether the video is worth viewing or not. In video abstraction, there are two approaches such as summary sequences and highlights. The summary sequences are good for documentaries because they give an overview of the entire video whereas highlights are good for movie trailers because they contain only the most interesting video clips [1].

To effectively construct video abstraction such as summary sequences and highlights, it is necessary to understand the video structure semantically. Previous research works are mainly done on the shot-level analysis using low-level features [1-7]. However, these approaches cannot well reflect semantics in the video abstraction process.

In order to reflect semantic features on video abstraction, it is necessary to analyze the scene-level features because a scene is a unit which has semantic coherences in video data. One possible approach to analyze scene-level features is to compute the contextual information within a scene or inter-scenes.

In this paper, we propose a novel scene-level analysis method for video abstraction by computing contextual information. This paper is organized as follows. Section 2 describes previous works on video abstraction. Section 3 discusses scene-level contextual information. We present local contextual information and global contextual information in detail. Section 4 shows experimental results using our approach.

2. Related Work

Many researchers have made great efforts in recent years on the video abstraction [1-7]. Most of research works are about constructing summary sequences from video data. Various domains of video data such as movies, TV dramas, documentaries, and newscasts are tested to create good quality of summary sequences.

For the abstraction of documentaries and newscasts in the CMU Informedia Digital Video Library project, video skimming [2] is proposed. In this approach, the video data and the transcript are aligned by the word spotting and the language analysis is used to identify important words in the transcripts. For example, if a face is detected in the video data and a proper name appears in the transcript, it is assumed that a person is introduced and therefore a 2-second video clip from this scene is selected into the summary sequences.

Hanjalic *et* al. [3] propose an automated video abstraction method by removing visual-content redundancy among video frames. They first group all frames of a video sequence into multiple clusters and then select the most suitable clustering options using an unsupervised procedure for cluster-validity analysis. From each valid cluster, key frames are extracted and then the video shots, to which key frames belong, are concatenated to form the summary sequences.

The generation methods of pictorial summary sequences are also proposed. Yeung *et* al. [4] propose a shot-based organization structures in which the story flow is shown by the scene-transition graph. Scene transition graph uses visual and temporal information and incorporates temporal relationships between the shots to build a graph. This approach is good for the efficient browsing and organization of video.

Uchihashi *et* al. [5] propose a method for creating pictorial video summaries like comic books. They compute the relative importance measure of each segment based on segment's rarity and duration. Weighted key frame selection method based on the knowledge of video contents is used. They test meeting videos to evaluate their method. Their system captures semantically important events with a compact arrangement of small images, and is suitable for web-based applications.

In contrast to summary sequences, a few works are done on the generation of the highlights from video. Lienhart *et* al. [1,6] propose an automated generation of movie trailers by investigating low-level visual and audio features, motion information, and color information. They use heuristics over the basic physical parameters of the video data to select clips of important objects, people, action, dialogs, title text, and title music. After that, extracted video clips, audio pieces, images, and texts are composed to make an abstract.

Babaguchi [7] proposes a method for abstracting sports video using event-based indexing scheme. To detect events in video data, he uses closed caption stream analysis and the visual stream analysis. After detecting events, he computes an impact factor reflecting the importance of the event and then appropriate highlights are selected. He tests his method to American football games. The generated video clips are still different from actual clips used in manually generated video highlights, but the results point toward a promising direction in video abstraction.

Even though these approaches are useful in constructing video abstracts of reasonable quality, some important scene-level features like contextual information in video are missing. In the next section, we will discuss a new method to compute scene-level contextual information for video abstraction.

3. Contextual Information in the Scene

To understand video data semantically, it is desirable to analyze scene-level features because a scene or a story unit consists of consecutive shots and has the semantic coherences. In order to analyze scene-level features, we compute the contextual information in video scenes. We divide the scene context into local scene context and global scene context. We define the local scene context as the context within the scene such as the shots' coherences and their interactions in the scene. We also define the global scene context as the context of inter-scenes such as the similarity between scenes and the relative importance of each scene in composing of video data. In this section, we will discuss contextual information in the scene.

3.1. Local Contextual Information

Since a scene is a collection of consecutive shots, the local context in the scene refers to the shots' coherences and their interactions in the scene. Using shots' coherences in the scene, we group the shots into a number of clusters and find the representative shot and dominant objects from each cluster. From shots' interactions, we find dialogues that are usually important in characterizing a scene. Figure 1 shows the local contextual information in the scene. The scene in Figure 1 consists of 4 clusters and has dialogues between shot "A" and the shot "B".

3.1.1. Coherence Detection

To detect coherences between shots in the scene, we apply k-means clustering algorithm on feature vectors for all key frames of shots in the scene. The feature vector of the key frame is computed from the color information. One problem in k-means clustering is to find an optimal number of clusters for the given data set in advance. To find the optimal number, we first classify all key frames into 1-to-N clusters, where N is a large number like the number of shots in the scene. Then, we use the cluster-validity analysis technique proposed in [3] to find an optimal number. From this technique, we find clusters that minimize intra-cluster distances while maximizing inter-cluster distances. The cluster separation measure is defined as

$$\rho(n) = \frac{1}{n}\sum_{i=1}^{n} \max_{1 \le j \le n \wedge i \ne j} \left(\frac{\varsigma_i + \varsigma_j}{\mu_{i,j}}\right) \qquad (1)$$

where

$$\varsigma_i = \frac{1}{E_i}\sum_{k=1}^{E_i}\left|\vec{\phi}(k|k\in i)-\vec{\phi}(c_i)\right| ,$$

$$\mu_{i,j} = \left|\vec{\phi}(c_j)-\vec{\phi}(c_i)\right| .$$

ς_i is the intra-cluster distance of cluster *i*, while μ_{ij} is the inter-cluster distance of cluster *i* and *j*. The optimal number of clusters is selected from the lowest of $\rho(n)$.

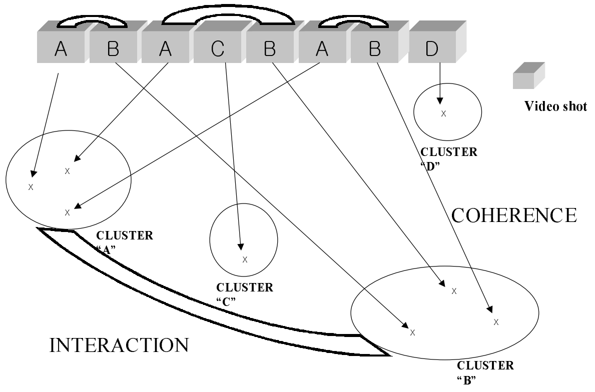


Figure 1: Local Contextual Information

From each cluster, a frame which is the closest to the center of the cluster is chosen as a representative frame for each cluster and the shot to which the representative frame belongs is the representative shot of the cluster. Since a scene consists of a number of clusters, we can extract several representative shots from each scene.

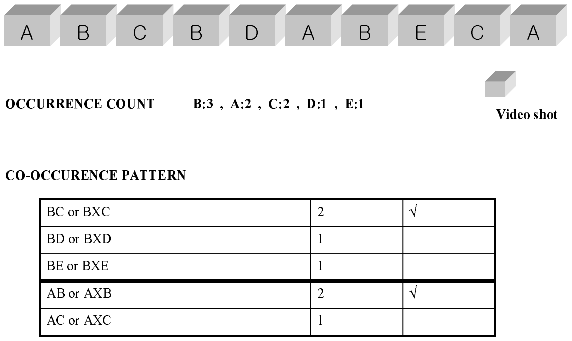
From the key frame of each representative shot, we separate foreground regions and background regions [8]. Then, we choose the largest foreground region in the frame and denote it as the dominant object of the shot. If the ratio of the foreground regions to the whole frame is less than 20 %, we ignore that frame in computing dominant objects. We also compute meaningful shot's camera motion phase and intensity using optical flow [9]. The camera motion in the shot is classified into "panning", "tilting", "zooming" and "no camera motion" cases [10]. So, by computing coherences between shots, we can detect representative shots in the scene. From the representative shots, we also find dominant objects and shots' camera motion in the scene. If the number of dominant objects is larger than one in the scene, it is desirable to detect interactions between objects.

3.1.2. Interaction Detection

Shots' interaction represents a dialogue between shots and is important in understanding video data. To detect dialogues, we first label each shot in the scene according to the clusters to which the shot belongs. After that, the scene is represented by a sequence of labels or symbols (see Figure 1). From the sequence of symbols, the dialogue detection finds desirable symbol patterns. One method to detect dialogues is to use periodic analysis transform and statistical tests [11]. With this approach, however, it is very difficult to detect dialogues in the complex scene like ABCBDABECA symbol sequence. In this sequence, the shots "A", "B" and "C" have interactions like dialogues.

In our approach, we detect dialogues by counting co-occurrence patterns *P*(*i, j*) in the sequence. We first count the occurrences for each symbol. Starting from the symbol having the largest occurrence count, we count the number of co-occurrence patterns *P*(*i, j*) with displacement *d* where *i* and *j* are a pair of symbols in the sequence, and *d* is the displacement between two symbols. In counting the number of co-occurrence patterns, we set *d* to 0 or 1. Suppose that there is a scene's symbol sequence like "ABCBDABECA". This is shown in Figure 2. "B" has the largest occurrence count in the symbol sequence. Then, we count the co-occurrence patterns *P*(*i, j*) where *i* is "B" and *j* can be "A", "C", "D" or "E". The displacement *d* is 0 or 1. For example, if *d* is 0, the possible co-occurrence patterns are "BA", "BC", "BD", or "BE". If *d* is 1, the possible co-occurrence pattern is "BXA", where the X represents the shot symbol in that position which can be ignored in counting the number of co-occurrence patterns. If the co-occurrence pattern occurs more than once, the shots representing co-occurrence pattern have interactions. In this example, shot "B" has the largest occurrence count of 3. So, we count the co-occurrence pattern *P*(*i, j*) where *i* is B and *j* can be other symbols except symbol "B" with *d* = 0 or 1. The co-occurrence pattern like "BC" or "BXC" has the count of 2. The interaction is checked as "√". So, shot "B" and shot "C" has dialogues. Next, we count another co-occurrence pattern *P*(*i, j*) where *i* is the shot having the second largest occurrence count (see Figure 2). In our example, shot "A" has the second largest occurrence count. After counting patterns, we find shot "A" and shot "B" have interactions. Therefore, shot "A", shot "B", and shot "C" have interactions in the scene. They may have dialogues.

Usually, the scenes in video can be classified into three types: a progressive scene, a dialogue scene, and a hybrid scene [11]. A progressive scene refers to the scene having linear progression of visuals without any repetitive structures. A dialogue scene, however, has repetitive structures. When a dialogue scene is embedded in a progressive scene, the resulting scene is a hybrid scene. The progressive scene is characterized by dominant objects and camera motions of the representative shots. The dialogue scene is characterized by shots' interaction and the hybrid scene is characterized by dominant objects, camera motion and shots' interaction. So, by computing local contextual information in the scene, we can understand the scene type and structure.



OCCURRENCE COUNT B:3 , A:2 , C:2 , D:1 , E:1

CO-OCCURENCE PATTERN

BC or BXC	2	√
BD or BXD	1	
BE or BXE	1	
AB or AXB	2	√
AC or AXC	1	

Figure 2: Dialogue detection method

3.2. Global Contextual Information

Global contextual information in the scene refers to the video scene's environment or its relationship with other video scenes. It is the context existing in the inter-scenes. In order to represent the global context of a given scene, we compute the scene's similarities and relative importance in comparison with other scenes. From the global contextual information, we can detect dominating scenes in the video according to the criteria. Figure 3 shows global contextual information in the scene.

3.2.1. Similarity

Similarity information of inter-scenes is important for video abstraction because it provides the semantic structure in understanding video data. The similarity amongst scenes is computed by applying hierarchical clustering algorithm on the key frames of representative shots in each scene. As dissimilarity measure, we use L^1 color histogram distances between two frames. For hierarchical clustering of key frames, we use a single-linkage algorithm [12]. This is obtained by defining the dissimilarity value between two clusters to be the smallest dissimilarity value between a possible pair of two key frames that belong to different clusters. The dissimilarity values between clusters are updated in the process of grouping.

After hierarchical clustering, a dendrogram is formed and we can find desirable video scene clusters by dissecting at the specific level. Finally, the scenes are grouped into a number of clusters (see Figure 3). From the cluster of scenes, we can compute various features for video abstraction.

3.2.2. Relative Importance

The relative importance measure of a given scene is another component in a scene's global context that we use. This is important in constructing video abstraction because it is necessary to make scenes ordered according to their importance. However, it is very difficult because each scene has various features. In our approach, we group scenes into a number of clusters using scenes' similarity with each other and then compute relative importance measure of each scene in the cluster. The criteria used in importance measure are scene's length, the number of dominant objects, and interactions that are computed from the local contextual information. A scene is important if it is long and if there are interactions between objects. The importance measure of the scene *i* (I_i) is computed as

$$I_i = \frac{L_i}{\sum_{j=1}^{C} L_j} + INT * NUM \qquad (2)$$

where L_i is the length of scene *i* and the summation gives the total length of all scenes in the cluster *C*. *INT* represents the interaction in the scene. If the interactions exist in the scene, the value is 1. Otherwise, the value is 0. *NUM* is the number of dominant objects. If the scene has interactions with a large number of objects, the importance value is large.

Based on the relative importance value, we select the scenes whose importance values are larger than the threshold T_d as dominating scenes. The dominating scenes are the basis in selecting meaningful shots to generate video abstraction. The candidates for meaningful shots are those that have long duration, high contrast, large activities, or interactions. Usually, the long duration or high contrast of a shot delivers its importance to the viewer. The shot which has large activities can be a part of the action and is also important to a viewer. The interacting shots are usually the dialog scenes which includes important message. So, global contextual information and local contextual information at the scene level are useful in semantic understanding of video data.

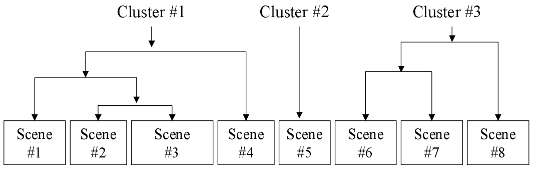


Figure 3: Hierarchical clustering

4. Experimental Results

We experiment our algorithm on two 20-minute TV dramas. To detect scene-level features, we use shot change detection algorithm based on the color histogram and region information. And then, one or a few key frames are extracted from each shot using color and motion information [10]. To select key frames of each shot, we detect the camera motion using optical flow in each video shot. Based on camera motion information, we classify the video shots into "panning", "tilting", "zooming", and "no camera motion". After that, in each category, we apply appropriate fuzzy rules to the selection of key frames based on dominant regions and their temporal variations in the shot [10].

We detect scene boundaries using continuous video coherence model based on short-term memory-based model [13]. Finally, the structure of video-scene-shot-key frame is extracted. In our experiment, the drama-1 has 26 scenes with 178 shots and drama-2 has 23 scenes with 172 shots.

To compute local contextual information in the scene, we apply k-means clustering algorithm on the key frames of shots for each scene. We find the optimal number of clusters using cluster-validity analysis technique [3]. As a feature vector in clustering, we use quantized YUV color histogram. In drama-1 case, the optimal number for clustering which is computed from cluster-validity analysis works well in 22 scenes out of 26 scenes. For each cluster, we choose the frame which is the closest to the cluster center as a representative frame. The shot to which the representative frame belongs is the representative shot of the cluster. For each scene, we detect several representative shots. This is shown in Figure 4.

To detect dialogues in the scene, we name the shots in the scene with a sequence of labels (or symbols) by assigning the same label for the shots in the same cluster. The dialogue detection is done by counting the number of co-occurrence patterns. Figure 5 shows an example. The label in the scene is sorted by the occurrence counts in descending order. From the first one, we detect co-occurrence patterns. In Figure 5, "L" has the largest occurrence count of 6. We compute the count of co-occurrence pattern *P(i,j)* where *i* is L and *j* is M or K with d = 0 or 1. The co-occurrence pattern "LM" or "LXM" occurs three times. Since it is larger than one, the interaction is checked as "√". So, shot "L" and shot "M" have interactions. Next, we detect co-occurrence patterns using "M" because "M" has the second largest occurrence count. We find that the shot "M" and the shot "L" have interactions. This dialogue detection algorithm has an accuracy of 92 % in two 20-minute dramas. Errors are usually generated from the incorrect clustering result.

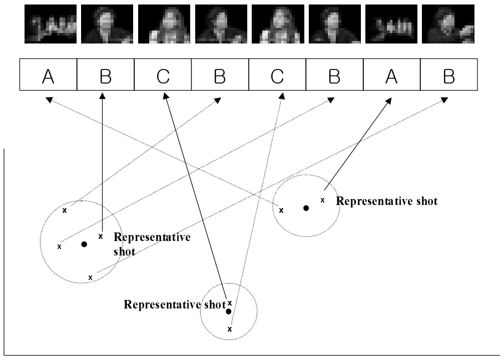
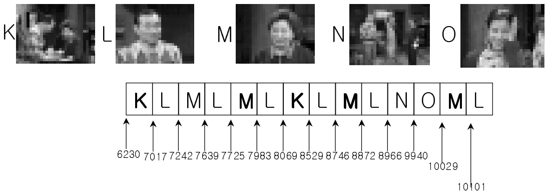


Figure 4: Clustering result in a scene

The global contextual information is computed by similarity and relative importance of scenes in video. To compute similarity between scenes, we perform hierarchical clustering on scenes and a dendrogram is formed. If the dissimilarity value in the clustering process is larger than threshold T_s, we stop clustering process. By doing this, the dendrogram is not formed. In drama-1, we obtain 7 clusters. For each cluster, the scenes' length, the number of dominant objects, and scene's motion are computed. We detect dominating scenes using relative importance measure as stated in Section 3.2.2. The dominating scenes have interactions, large number of dominating objects and long duration. They are the basis in the selection of meaningful shots for constructing video abstracts. Therefore, the contextual information at the scene-level is important for video abstraction process.



OCCURRENCE COUNT **L:6 , M:4 , K:2 , N:1 , O:1**

CO-OCCURENCE PATTERN

LM or LXM	3	√
LK or LXK	1	
LN or LXN	1	
ML or MXL	4	√
MK or MXK	0	

Figure 5: An example of dialogue detection

5. Conclusion

In this paper, we discuss a new scene-level analysis method for video abstraction. To understand video data semantically, we compute local and global contextual information at the scene-level. We define the local scene context as the context within the scene such as the shots' coherences and their interactions in the scene. We also define the global scene context as the context of inter-scenes such as the similarity and relative importance of the scenes in composing video data. Using contextual information, dominating scenes can be selected for video abstraction. From dominating scenes, meaningful shots are selected to construct video abstracts.

There are still some limitations to practically realize video abstraction procedure. The first limitation is that it is very difficult to develop a system which captures meaningful shots automatically because meaningful shots are decided by the user's perception or affection. This is due to the fact that the meaningfulness in human beings is very subjective. So, it is not an easy task to make an automated abstraction process such that similar abstraction results are generated both manually and automatically. Another limitation is that the evaluation procedure is also subjective. That is, there is no objective ground truth for the evaluation.

6. Acknowledgements

This work is funded by University Research Program supported by Ministry of Information & Communication in Republic of Korea.

7. References

[1] R. Lienhart, S. Pfeiffer, and V. Effelsberg, "Video Abstracting," *Comm. of ACM*, Vol. 40, No. 12, pp. 55-62, Dec. 1997.

[2] M. Christal, M. Smith, C. Taylor and D. Winkler, "Evolving Video Skims into Useful Multimedia Abstractions," *Proc. CHI'98,* pp. 171-178, 1998.

[3] A. Hanjalic and H. Zhang, "An Integrated Scheme for Automated Video Abstraction Based on Unsupervised Cluster-Validity Analysis," *IEEE Trans. Cir. & Sys. for Video Tech*., Vol. 9, No. 8, pp. 1280-1289, Dec. 1999.

[4] M. Yeung, B. Yeo and B. Liu, "Segmentation of Video by Clustering and Graph Analysis," *Computer Vision and Image Understanding,* Vol. 71, No. 1, pp. 94-109, 1998.

[5] S. Uchihashi, J. Foote, A. Girgenshon, and J. Boreczky, "Video Manga: Generating Semantically Meaningful Video Summaries," *Proc. ACM MM'99*, 1999.

[6] S. Peiffer, R. Lienhart, S. Fisher, and W. Effelsberg, "Abstracting Digital Movies Automatically, " *Int. Jour. Visual Communication and Image Representation,* Vol. 7, No. 4, pp. 345-353, 1996.

[7] N. Babaguchi, "Towards Abstracting Sports Video by Highlights," *Proc. ICME'00*, New York, NY, Aug. 2000.

[8] M. Kim, J. Choi, D. Kim, H. Lee, C. Ahn and Y. Ho, "A VOP Generation Tool: Automatic Segmenation of Moving Objects in Image Sequences Based on Spatio-Temporal Information," *IEEE Trans. Cir. Sys. for Video Tech*., Vol. 9, No. 8, pp. 1216-1226, Dec. 1999.

[9] B. D. Lucas and T. Kanade, "An Iterative Technique of Image Registration and Its Application to Stereo," *Proc. IJACI,* pp. 674-679, 1981.

[10] H. –B. Kang, "Key Frame Selection using Region Information and Its Temporal Variations," *Proc. IMSA'99,* Nassau, Grand Bahamas, pp. 33-37, Oct. 1999.

[11] H. Sundaram and S. Chang, "Determining Computable Scenes in Films and Their Structures using Audio-Visual Memory Models," *Proc. ACM MM'00*, 2000.

[12] E. Grose, R. Johnsonbaugh and S. Jost, *Pattern Recognition and Image Analysis*, Prentice Hall PTR, 1996.

[13] H. –B. Kang, "Continuous Video Coherence Computing Model for Detecting Scene Boundaries," *SPIE Proc. Internet Multimedia Management Systems II*, Denver, CO, pp. 1-9, Aug. 2001.