Effective Video Content Abstraction by Similar Shots Clustering

Shouqun Liu Ming Zhu Quan Zheng
Department of Automation, University of Science and Technology of China
Hefei, Anhui, 230027, P.R.China
E-mail: sqliu@mail.ustc.edu.cn

Abstract

With the widespread use of digital video technology and the explosion of video data, effectively and efficiently organizing the video content becomes an important issue. In past years, various methods and techniques have been proposed towards this problem, among which organizing video data by summarizing the contents is a practical and useful method. The purpose of this article is to provide a hierarchical video abstraction strategy based on similar video shots clustering. The abstraction method treats a video as a set of shot clusters, and the video abstraction is composed of the exemplar shots of these clusters. The effectiveness of the proposed method is proved by experiments.

1. Introduction

In recent years, multimedia applications have been rapidly developed and widely deployed with the advance of computer networks and hardware. Due to the widespread use of digital video system, the amount of video data becomes huge consequently. Effectively organizing the video data while providing the end-user a friendly way to browse the video content turns to be a crucial problem. There are many techniques to solve this problem; video abstraction is a generally used method among them. Video abstraction [1] [2] provides the users a partial perspective of an entire video, which allows user to browse the video without fully watching it. Video abstraction techniques are mainly designed to facilitate browsing of large video collections, and they also help the user to navigate and interact with one single video sequence in a nonlinear manner analogous to the video editing storyboard. It is important for reducing network bandwidth cost and browsing time especially under online video service circumstances.

To date, a lot of research works have been done to find the best solutions for effectively generating video abstractions. The developed techniques target video data range from various domains, including movies, documentaries, sports, news, etc. and results that have been reported are promising. However, video abstraction is still largely in the research phase; applications are still limited in both complexity of method and scale of deployment.

The literature in video abstraction can be categorized into two classes: content abridgement and event summary. In this paper, we propose a shot level clustering method for video content abstraction. The method is not only limited to video abstraction, but also can be applied for such tasks as video indexing, multimedia mining, etc.

The reminder of this paper is organized as follows: Section 2 describes an over view of the shot level abstraction algorithm. Section 3 discusses the shot based clustering algorithm. Interactive browsing technique is presented in section 4. Section 5 gives the experiments and results. Finally, we conclude the paper with a summary.

2. Video Abstraction by Shots Clustering

2.1. System overview

In this section, we describe the overview of shot clustering based video abstraction system. Figure.1 shows the procedure of the proposed abstraction approach. First we segment the input video into independent shots [3] [4]; the visual feature of each shot will be extracted after that. With the features, the shots are clustered into content correlated shot clusters according to similarities between them; the video abstractions are generated according to the clustering result and stored into video database. The end users can browse videos by generated abstractions, they can also interact with the video browsing system and browse video in a coarse-to-fine way.

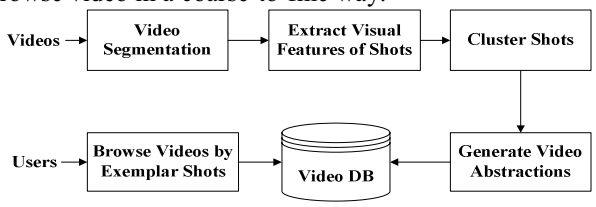

Figure 1. The procedure of clustering based video abstraction system

Shot clustering method and abstraction generation strategies are the emphases, so the rest of this paper

978-1-4244-2179-4/08/$25.00 ©2008 IEEE

mainly discusses them in detail.

2.2. Visual features and similarity between shots

For all segmented shots, visual features are needed to be extracted. With the visual features, the similarities between each other are calculated consequently before clustering.

We use the maximum matching based similarity measure that employs the color information of shot frames. This measure was originally proposed in [5] by Rasheed and Shah, and was improved by Peng and Ngo in [6].

Given two shots $\boldsymbol{X}$ with p frames and $\boldsymbol{Y}$ with q frames, let $\boldsymbol{x}_i$ denotes the i-th frame in $\boldsymbol{X}$ and $\boldsymbol{y}_j$ denotes the j-th frame in $\boldsymbol{Y}$; the similarity between $\boldsymbol{X}$ and $\boldsymbol{Y}$ is defined as follows:

$$Similarity(X,Y) = Sim_{color}(X,Y) \tag{1}$$

where the $Sim_{color}(X,Y)$ is the color similarity between $\boldsymbol{X}$ and $\boldsymbol{Y}$, and the maximum matching is defined as:

$$Sim_{color}(X,Y) = \max_{i\in p, j\in q}(Sim_{color}(x_i, y_j)) \tag{2}$$

$Sim_{color}(x_i,y_j)$ is the color similarity between the i-th frame in $\boldsymbol{X}$ and the j-th frame in $\boldsymbol{Y}$. HSV color histogram is employed here to measure similarity between frames.

$$Sim_{color}(x_i, y_j) = \frac{\sum_{h,s,v}\min(H_i(h,s,v), H_j(h,s,v))}{\min(\sum_{h,s,v}H_i(h,s,v), \sum_{h,s,v}H_j(h,s,v))} \tag{3}$$

We discard the motion information in the similarity calculation because the color information is sufficient. According to [5] and [6], the measure is effective for most cases in shot similarity measurement.

3. Clustering with Affinity Propagation

3.1. Affinity propagation

Clustering based on similarity measure is a crucial problem. Recently, a powerful algorithm named affinity propagation (AP) based on message passing between data points was proposed by Frey and Dueck [7]. The algorithm achieved a considerable improvement over the standard clustering methods such as k-means, k-medoids and spectral clustering.

Based on pair-wise similarities between data points, AP seeks to identify each cluster by one of its elements, the so-called exemplar. Each point in the cluster refers to this exemplar, and each exemplar is required to refer to itself as a self-exemplar. This hard constraint forces clusters to appear as starts of radius one: there is one central node, and all other nodes are directly connected to it. Subject to this constraint, AP seeks to maximize the overall similarity of all data points to their exemplars. The solution is approximated following the ideas of belief-propagation.

There are two kinds of messages passed in AP between data points: the responsibility and the availability, each of which takes a different kind of competition into count.

The pair-wise similarity between $\boldsymbol{i}$ and $\boldsymbol{k}$ is defined as $s(\boldsymbol{i},\boldsymbol{k})$. All the availability for each node is set to zero at the beginning: $\boldsymbol{a}(\boldsymbol{i},\boldsymbol{k})=0$, the responsibility $\boldsymbol{r}(\boldsymbol{i},\boldsymbol{k})$, sent from data point $\boldsymbol{i}$ to the candidate exemplar $\boldsymbol{k}$, reflects the accumulated evidence for how-well suited for $\boldsymbol{k}$ is to be chosen as a exemplar for point $\boldsymbol{i}$, comparing with other potential exemplars for point $\boldsymbol{i}$. The responsibility of $\boldsymbol{k}$ for $\boldsymbol{i}$ in the n+1 iteration step is computed as:

$$\begin{cases} r_{n+1}(i,k) = \lambda r_n(i,k) + (1-\lambda)\Delta r_n(i,k) \\ \Delta r_n(i,k) = s(i,k) - \max_{k'\neq k}\{a_n(i,k') + s(i,k')\} \end{cases} \tag{4}$$

where $\boldsymbol{s}(\boldsymbol{i},\boldsymbol{k})$ is the similarity between point $\boldsymbol{i}$ and $\boldsymbol{k}$. For $\boldsymbol{k}=\boldsymbol{i}$, the self-responsibility reflects the accumulated evidence that point $\boldsymbol{k}$ is selected as an exemplar, based on its input preference tempered by how ill-suited it is to be assigned to another exemplar.

The availability $\boldsymbol{a}(\boldsymbol{i},\boldsymbol{k})$, sent from the candidate exemplar point $\boldsymbol{k}$ to point $\boldsymbol{i}$, reflects the accumulated evidence for how appropriate it would be the exemplar for $\boldsymbol{i}$. The availability update gathers evidence from data points as to whether each candidate exemplar would make a good exemplar, whereas the responsibility update lets all candidate exemplars compete for ownership of a data point. The n+1 step of availability is:

$$\begin{cases} a_{n+1}(i,k) = \lambda a_n(i,k) + (1-\lambda)\Delta a_n(i,k) \\ \Delta a_n(i,k) = \min\{0, r_n(k,k) + \sum_{i'\notin\{i,k\}}\max\{0, r_n(i',k)\}\} \end{cases} \tag{5}$$

For $\boldsymbol{i}=\boldsymbol{k}$, the update of self availability $\boldsymbol{a}(\boldsymbol{k},\boldsymbol{k})$ is in a different way:

$$\begin{cases} a_{n+1}(k,k) = \lambda a_n(k,k) + (1-\lambda)\Delta a_n(k,k) \\ \Delta a_n(k,k) = \max_{i'\neq k}\{0, r_n(i'k)\} \end{cases} \tag{6}$$

This message reflects accumulated evidence that point $\boldsymbol{k}$ is an exemplar, based on the positive responsibility sent to candidate exemplar $\boldsymbol{k}$ from other points.

The λ is a damping factor and is set to 0.5 usually.

After the convergence, availability and responsibility are combined to identity exemplars. For point $\boldsymbol{i}$, its corresponding exemplar is obtained as:

$$E = \boldsymbol{a}(\boldsymbol{i},\boldsymbol{k}) + \boldsymbol{r}(\boldsymbol{i},\boldsymbol{k}) \tag{7}$$

For point $\boldsymbol{i}$, the value of $\boldsymbol{k}$ that maximizes $\boldsymbol{E}$ either identifies point $\boldsymbol{i}$ as an exemplar if $\boldsymbol{k}=\boldsymbol{i}$, or identifies that the data point $\boldsymbol{k}$ is the exemplar for point $\boldsymbol{i}$. After the algorithm convergence, cluster exemplars can be determined and data points are assigned to their exemplars.

3.2. Shot clustering with affinity propagation

Each shot is treated as a data point during the AP clustering. The pair-wise similarity measure given in Section2.2 is used here as similarity evaluation for shots.

The self-similarity $\boldsymbol{s}(\boldsymbol{k},\boldsymbol{k})$, which is called the preference, affects the likelihood of shot $\boldsymbol{k}$ to be chosen as a exemplar. Typically it is set to be the median of the input similarities. Based on the heuristic rule that shot

with longer duration is more likely to be an exemplar, we define the preference value $s(k,k)$ as:

$$s(k,k) = \frac{t_k}{\max_{i=1\ldots n}(t_i)} median_{i \neq k} s(i,k) \quad (8)$$

where t_i is the duration of i-th shot, and $median_{i \neq k} s(i,k)$ is the median of input similarities.

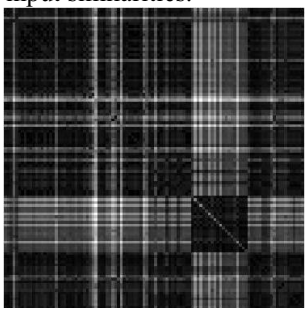

Figure 2. The pair-wise similarity matrix of a sample video with 161 shots, with the clustering algorithm, the shots is clustered into 19 clusters.

Computing all pair-wise similarities and updating every responsibility and availability have a considerable computation when the number of input data is huge. Sparse clustering is used to reduce computation cost. The sparse clustering only calculates top-k nearest neighbor (k-NN) similarities for every shot, while updating of responsibility and availability is confined in the connected set.

3.3. Abstraction generation by exemplar shots selection

The exemplar shots are used to represent each shot clusters. They will also be served as abstractions of input video. So it is important to select appropriate exemplars. Traditional methods evolve a post processing procedure of select representatives [1].

Different from other clustering algorithms such as k-means and k-medoids, the cluster centers in AP are actually real data points in the data clusters, which are the so called exemplar data points. In general, the principal of AP algorithm is the process of detecting such exemplar data points; this feature is an advantage of Affinity Propagation. As a result, the exemplar shots are naturally the exemplar data points after the AP algorithm convergence.

3.4. Hierarchical shot clustering

For multiple video sources, clustering all the shots together is a computational cost task and requires a large amount computing time. To handle this problem, we propose a hierarchical shot clustering scheme: shots in each video are clustered separately, after obtaining all the exemplars of each video, the AP algorithm will be applied to the exemplar shots again, and then the exemplar shots generate high layer exemplars.

The hierarchical shot clustering can also progress under distributed computing environment.

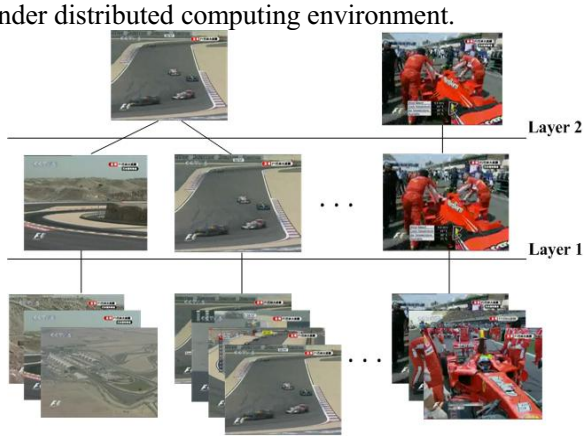

Figure 3. Hierarchical clustering of shots

4. Interactive Navigation of Videos

The main purpose of video abstraction is to facilitate browsing of a video database. Our shot clustering based abstraction also allows users to navigate and interact with one single video sequence in a nonlinear manner analogous to video editing storyboard.

4.1. Ranking of exemplar shots

The clustering result is ranked for effective browsing, which is the same as what most text search engines do. The factors that influences the ranking results include cluster size (R_C) and duration of exemplar shot (R_T).

The cluster size contains the information that how many shots the exemplar shot represents. The larger the cluster size, the exemplar is ranked more important; the rank value also increases with the duration of exemplar shot. We define a ranking criteria $R(i)$ for the i-th exemplar:

$$R(i) = w_c \frac{R_C(i)}{C} + w_T \frac{R_T(i)}{T} \quad (9)$$

where $R(i)$ is a normalization formula in which the C denotes the total number of shots and T is the total length of the video. w_c and w_t are weight factors and typically set as w_c=w_t.

4.2. Interactively browsing

The video abstractions are stored in video database, and the shots are ranked according the rule in Section4.1. Our system allows users to browse video content in an interactive way; the browsing flow is just like navigating in the web. Since all exemplar shots are representative of their clusters and linked with other

shots in the cluster, users may start browsing from the exemplar shots; they can follow the links between the exemplar shot and the shots the exemplar represents to deeply browse similar shots.

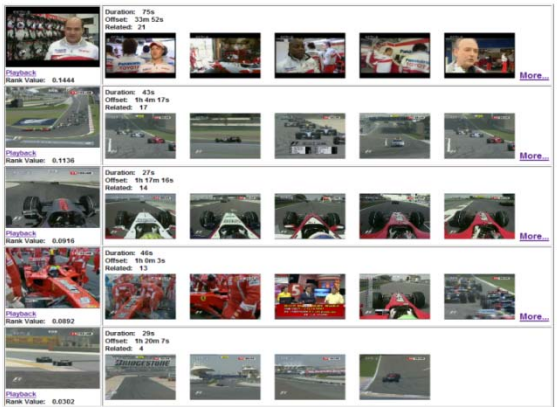

Figure 4. The user interface for browsing, exemplar shots are ranked according to their ranking value. The browsing process is a coarse-to-fine navigation that can speed up the users to access video content.

5. Experiment

We now describe the experiments used to empirically demonstrate the effectiveness of our clustering approach; the experiments are conducted on a set of videos. The datasets chosen are Formula.1 match videos of season 2007. We recorded five matches with a total duration of 10 hours. These videos are first segmented into independent shots.

To evaluate the quality of shot clustering, purity (P) [8] is used as a measurement. For a given set of shots belongs to c distinctive groups, suppose that they are clustered into m clusters C_j, j=1…m, and for a single cluster C_j, purity is defined as:

$$P(C_j) = \frac{1}{|C_j|} \max_{k=1\ldots c} |C_{j,k}| \tag{10}$$

where $|C_j|$ is the size of cluster, and $C_{j,k}$ is a set of shots in cluster C_j that belongs to the ground truth group k. The *average purity* is:

$$P = \frac{1}{c} \sum_{k=1}^{c} \left(\frac{1}{m_k} \sum_{j=1}^{m_k} P(C_j) \right) \tag{11}$$

where m_k denotes the number of clusters containing shots from ground truth group k, and c is the number of ground truth group.

Table 1. Clustering results for F1 video dataset

Video	#Shots	*maxC*	*minC*	*c*	*k*	*P*
1.Australia	641	53	4	36	43	0.93
2.Malaysia	774	62	7	35	54	0.88
3.Bahrain	548	81	3	41	44	0.91
4.Canada	665	47	11	38	39	0.94
5.Monaco	977	42	6	47	52	0.84

Table.1 shows the result of clustering, where *maxC* is the maximum cluster size and *minC* is minimum cluster size.

According the experiment results, our clustering method achieves a good performance. Because our method uses a maximum matching method for shot similarity evaluation, the similarity calculation consumes a lot of time. However the time for clustering is acceptable. Other features such as SIFT feature can also be used to evaluate shot similarity.

6. Conclusion

Based on affinity propagation, this paper proposes a novel video abstraction strategy by similar shot clustering. Given an input video, segmentation module first segments it into shots. Then the clustering algorithm clusters similar shots and selects appropriate shots as exemplar shots. The exemplar shots compose of abstraction of the video. In order to facilitate the users to browse, ranking scheme and interactive user interface are also studied in this paper. Experiments have demonstrated that the clustering based method is effective.

Acknowledgements

The authors would like to thank Network Communication System & Control lab for supplying experiment environment. They also thank the anonymous reviewers. The research is supported by the National High-Tech Research and Development Program of China (863) (No. 2008AA01Z408)

References

[1] B.T. Truong and S. Venkatesh. “Video abstraction: A systematic review and classification”. ACM Trans. Multimedia Computing, Communications, and Applications, 2007, 3(1)

[2] W. Tavanapong and J.Y. Zhou, "Shot Clustering Techniques for Story Browsing", IEEE Trans on Multimedia, 2004. 6(4):517-527

[3] Yuan J H, Li J M. “A Unified Shot Boundary Detection Framework Based on Graph Partition Model”. Proc. of the 13th annual ACM international conference on Multimedia (MM), 2005. p539-542

[4] Jinhui Yuan, Huiyi Wang, Lan Xiao. “A Formal Study of Shot Boundary Detection” [J]. IEEE Trans. Circuits and Systems for Video Technology, 2007, 17(2):168-186.

[5] Zeeshan Rasheed and Mubarak Shah, “Detection and Representation of Scenes in Videos”. IEEE Trans. Multimedia, 2005,7(6):1097-1105

[6] Peng Y, Chong-Wah Ngo and Xiao J. “OM-based video shot retrieval by one-to-one matching”. Multimedia Tools and Applications, 2007, 34(2):249-266

[7] B. J. Frey and D. Dueck. “Clustering by passing messages between data points”. *Science*, 2007, 315:972–976

[8] Y. Chen, J. Z. Wang, R. Krovetz. “CLUE: cluster-based retrieval of images by unsupervised learning”. IEEE Trans. Image Processing, 2005, 14(8):1187-1201.