INTERACTIVE CLUSTERING OF VIDEO SEGMENTS FOR MEDIA STRUCTURING

Y. Kinoshita, N. Nitta, and N. Babaguchi

Graduate School of Engineering, Osaka University
2–1 Yamadaoka, Suita, Osaka, 565–0871 Japan
{yukihiro, naoko, babaguchi}@nanase.comm.eng.osaka-u.ac.jp

ABSTRACT

Structuring video data is necessary for its effective retrieval and summarization. In particular, collecting similar scenes from semantic aspects highly contributes to the structuring. In this paper, we propose a method of clustering the scenes with relevance feedback, which may be able to bridge the gap between the video data and its semantics. First, spatio-temporal video segments of a fixed length are clustered according to image features of each segment. Then, a user performs feedback to the results of clustering, whether each segment is relevant to the cluster it belongs to. The clustering accuracy can be improved through the interaction based on the feedback information. For diverse kinds of video streams, we investigated how the feedback should be given and demonstrated the effectiveness of the interactive clustering.

1. INTRODUCTION

The amount of data offered through televisions or the Internet is increasing due to recent progress of communication technology, and so is the time and effort required to retrieve particular video scenes from the video stream. In order to use the video data efficiently and effectively, media structuring has become indispensable. In media structuring, semantic-based handling should be taken into account.

One way of media structuring is clustering where the video stream is divided into video segments, and similar video segments are put together and assigned a semantic label. Usually, video segments of variable length, called shots, are used as a unit of video segments. However, there is currently no technique which is able to detect shots with complete accuracy. Therefore, in order to exempt the need for the perfect shot detection, our method uses fixed-length video segments called video packets[1].

Let us mention related work. There is another clustering method where short continuous frame sequences called video scenelets[2] are used. Although their and our methods are similar in terms of not taking shot changes into consideration, the features are different. H.Lu et al.[2] employed the HSV color histogram computed as the average of all the frame color histograms in the scenelet. Since video streams have temporal features, temporal analysis of video streams is also important. Therefore, we employ tensor histograms[3] for spatio-temporal images to analyze temporal features as well as color histograms. Ngo et al.[3] demonstrated that tensor histograms were effective for temporal analysis.

Since the system performs clustering with low-level image features such as color and texture, there is a semantic gap between the clustering results and users' interpretation. To bridge the semantic gap, we apply relevance feedback[4, 5, 6] to learn the semantic concept from users. Relevance feedback is a tool where a system presents more suitable results based on the feedback information whether the results are relevant to the user's queries. Rui et al.[4] proposed a method of weight updating for feature vectors based on the user's feedback, and our method is deeply related with their method. While the conventional methods[4, 5, 6] applied feedback to cluster still images, our method applies feedback to cluster video segments.

In this paper, we show how the accuracy of clustering changes with relevance feedback. We then investigate which features work effectively for different types of videos by changing the combination of the tensor and color histograms when extracting features. We also investigate several ways to give feedback in order to reduce users' efforts without degrading the clustering accuracy.

2. OUTLINE OF THE PROPOSED METHOD

Clustering of video segments is to put similar video segments together based on their feature vectors. Our method uses fixed-length video packets instead of shots as a unit of video segments for clustering. Video packets are defined as partial video streams of a fixed length, while shots are defined as consecutive image frames taken by a single camera.

The flow of our proposed method is shown in Fig.1. First, we divide the video stream into the video packets. The spatio-temporal images or frames are obtained from each video packet. Clustering is performed based on the feature vectors calculated from each spatio-temporal image

0-7803-9332-5/05/$20.00 ©2005 IEEE

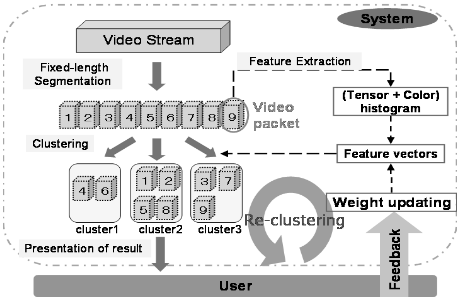

Fig. 1. The Flow of the clustering.

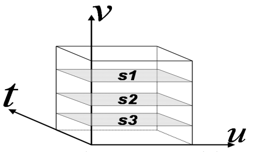

Fig. 2. Spatio-temporal images.

or frame using tensor histograms[3] and color histograms. When the results of clustering are shown to a user through an interface, the user specifies whether each video packet is relevant or irrelevant to the clusters in order to refine the effect of video packets that induced the incorrect clustering results. Based on the feedback information from the user, the system updates the feature vectors and generates new feature vectors. The system performs re-clustering with the new feature vectors and presents the results of re-clustering to the user. We aim at improving the clustering accuracy by repeating the operations above.

3. CLUSTERING OF VIDEO PACKETS

In this section, we explain the details of each step of our method; feature extraction and clustering based on relevance feedback, and then alternative feedback operations to improve the efficiency of our system.

3.1. Feature Extraction

Features of video packets are extracted from spatio-temporal images or frames. Fig.2 shows a video packet, where v and u are the height and width of a frame respectively and t represents the time, and three spatio-temporal images are obtained horizontally with u and t. Since many types of camera work and movements of objects are horizontal, we obtain three spatio-temporal images (s1, s2, and s3 in Fig.2).

1) Tensor Histogram: One of the features of the spatio-temporal images is represented with tensor histograms[3]. Tensor histograms encode the distribution of local orientation in spatio-temporal images. Tensor histogram $\mathcal{M}(k)$ is computed from one spatio-temporal image, where $k = \{1, 2, \cdots, 8\}$ represents the quantized level. Our method uses the tensor histograms $(\mathcal{M}_{s1}(k),\mathcal{M}_{s2}(k),\mathcal{M}_{s3}(k))$ of three horizontal slices (s1, s2, and s3 in Fig.2) to generate feature vectors. As a result, the feature vector T is expressed as $T = (\mathcal{M}_{s1}(1),\cdots,\mathcal{M}_{s1}(8),\mathcal{M}_{s2}(1),\cdots,\mathcal{M}_{s2}(8),\mathcal{M}_{s3}(1),\cdots,\mathcal{M}_{s3}(8))$.

2) Color Histogram: Color features are also extracted from spatio-temporal images or frames. Our method uses RGB color histograms as color features. The color histogram is computed as the average of three spatio-temporal images or three frames of a video packet(the first, the middle, and the last frames), where the RGB color coordinates are quantized into 8(R), 8(G) and 8(B) bins. The color feature C is expressed as $C{=}(R(1),\cdots,R(8),G(1),\cdots,G(8),B(1),\cdots,B(8))$.

The system obtains one feature vector F by combining $T = (t_1, \cdots, t_{24})$ and $C = (c_1, \cdots, c_{24})$. F is expressed as $F = (t_1, \cdots, t_{24}, c_1, \cdots, c_{24})$.

3.2. Clustering based on relevance feedback

The system operates the K-means clustering using the set of feature vector F. The K-means algorithm is the most frequently used algorithm due to its simplicity and efficiency. The user gives his/her feedback whether each video packet is relevant or not to each cluster it belongs to after referring to the results of clustering presented by the system. The system updates the feature vectors in each cluster based on the feedback and operates the clustering with the updated feature vectors. The feature vectors are updated based on the feedback information as described below.

For the set of feature vectors F of video packets, the user specifies whether each video pakcet is relevant or not. For the set of relevant feature vectors, when the standard deviation of the i-th $(1 \leq i \leq 48)$ component of the feature vectors is large, the system recognizes that the user considers the i-th component is not important and gives it a small weight w_i. On the contrary, the system gives large weights to the components whose standard deviations are small. Based on the idea above, the weight w_i is defined as $w_i = 1/\sigma_i(1 \leq i \leq 48)$ where σ_i is the standard deviation of the i-th component. The system normalizes W_i with the sum of w_i $(W_i = w_i/\sum w_i)$. The system updates the feature vectors with W_i. When the feature vector is $F = (f_1, f_2, \cdots, f_{48})$, the updated feature vector is $F' = (W_1 f_1, \cdots, W_{48} f_{48})$. New feature vectors are generated by updating original feature vectors, and the new feature vectors are re-clustered with the K-means algorithm.

3.3. Improvement with Alternative Feedback Operations

Since relevance feedback is usually given to all packets, too much effort is required to deal with large video libraries. For example, a 2-hours video consists of 3600 2-seconds packets, which require 3600 user interaction per re-clustering cycle. To solve this problem, we propose two alternative ways to give the relevance feedback only to selected packets. Before giving feedback, we sort the packets in each

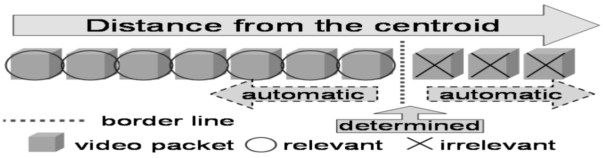

Fig. 3. Feedback by setting a border line.

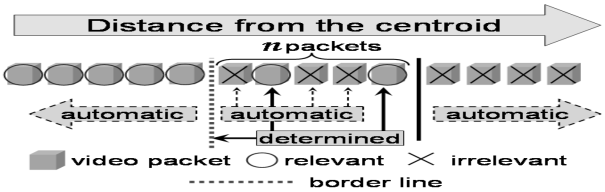

Fig. 4. Setting a border line and giving feedback only to relevant packets among n packets after the border line.

cluster sequentially from the closest one to the centroid of each cluster.

One feedback operation is to put a border line in each cluster as shown in Fig.3. A border line represents the threshold between relevant and irrelevant packets. The packets before the border line are determined as relevant, while other packets are determined as irrelevant automatically. The other feedback operation is to put a border line and give the feedback only to the relevant packets among n packets after the border line as shown in Fig.4. The other packets are determined as irrelevant automatically. It is considered that the number of n should be increased according to the length of the video stream. For aforementioned 2-hours video, these two feedback operations can respectively reduce the user's interactions from 3600 to *the number of clusters* or $n\times$*the number of clusters* per re-clustering cycle. The effectiveness for these two types of feedback operation is shown in the experimental results.

4. EXPERIMENTAL RESULTS

In this section, we present experimental results with various kinds of videos, all of which are 5-minutes long.

4.1. Example of Operation

Fig.5 shows a cluster before any feedback is given and how the user gave feedback to the cluster. The first frame of each video packet is presented as the representative image. Fig.6 shows the new cluster obtained by re-clustering based on the user's feedback information. Fig.5 and Fig.6 indicate that the clustering accuracy was improved with a single feedback operation. Furthermore, after the 10th re-clustering, the new cluster was generated as shown in Fig.7. We demonstrated that interactive re-clustering was able to improve the performance.

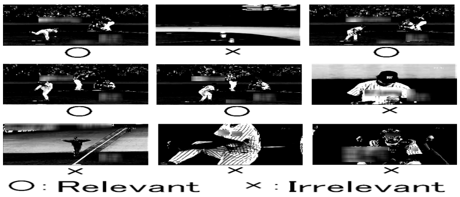

Fig. 5. Result of clustering (Initial cluster).

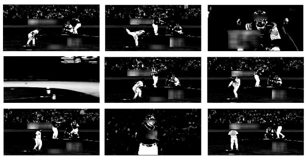

Fig. 6. Result of clustering (After the first re-clustering).

Fig. 7. Result of clustering (After the 10th re-clustering).

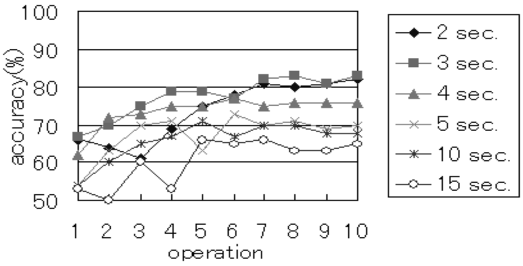

Fig. 8. Accuracy changes according to the packet length.

4.2. Experiments about Length of Video Packets

We examined how the clustering accuracy changed according to the length of the packets. Each video is composed of some categories. We determined categories for each video beforehand. The examples of the categories for baseball videos are the scenes of pitching, close-up of a player, etc. The user specifies if each element in the clusters is relevant based on the pre-defined categories. The results were evaluated with the accuracy (= the number of relevant packets / the number of all packets). The length of the packets was set to 2, 3, 4, 5, 10, and 15 seconds, and we investigated the clustering accuracy with three baseball videos using tensor histograms as features of video packets.

The results are shown in Fig.8. Fig.8 shows that when the length of the packets was too long, the recursive feedback did not have any effect on the performance. This is due to the camera work and shot changes in the videos. Since the packets with shot changes have different features from those without shot changes, the packets with shot changes can not be classified correctly. Therefore, the length of the packets should be set shorter for the videos with many shot changes. Moreover, the accuracy sometimes degraded after an iteration as shown in Fig.8. Since weight updating is performed within one cluster, the system does not learn the relation among clusters. Therefore, improving the accuracy of one cluster sometimes resulted in degrading the accuracy of other clusters, thereby degrading the total accuracy. Investigating a way to update feature vectors which reflects the relation among clusters will be one of our future work.

Table 1. Clustering accuracy using only tensor histograms (Packet length: 2 sec.)

video	kind of video	accuracy(%)	
		Initial	10th
1	Baseball	66	81
2	Sumo	70	89
3	Volleyball	70	89
4	Soccer	54	84
5	American football	48	92
6	Variety	59	80
7	Cooking	47	64
8	News	41	71
average		56.8	81.2

Table 2. Clustering accuracy using tensor histograms and color histograms of spatio-temporal images (Packet length: 2 sec.)

video	kind of video	accuracy(%)	
		Initial	10th
1	Baseball	65	81
2	Sumo	72	90
3	Volleyball	65	84
4	Soccer	60	76
5	American football	58	90
6	Variety	58	81
7	Cooking	48	66
8	News	43	70
average		58.6	79.8

Table 3. Clustering accuracy using tensor histograms of spatio-temporal images and color histograms of frames (Packet length: 2 sec.)

video	kind of video	accuracy(%)	
		Initial	10th
1	Baseball	66	86
2	Sumo	72	92
3	Volleyball	56	86
4	Soccer	54	79
5	American football	56	93
6	Variety	59	78
7	Cooking	64	85
8	News	50	69
average		59.6	83.5

4.3. Experiments with Feature Selection

We also applied our method to other types of videos (volleyball, news, etc.) with setting the length of the packets to 2 seconds. First, we only used the tensor histograms as feature vectors. The results are shown in Table 1. For eight videos, the initial accuracy was about 57% on average. After the 10th feedback, the accuracy was improved to about 81% on average. The accuracy for video 7 and 8 did not reach comparable levels by repeating the feedback. This results show that tensor histograms worked well for sports videos, which have a lot of motion.

Secondly, we used the tensor and the color histograms of spatio-temporal images as feature vectors. The results are shown in Table 2. The accuracy after the 10th feedback for each video hardly changed compared to the results in Table 1 and Table 2. The results in Table 2 show that the color histograms of spatio-temporal images did not bring further useful information.

Next, we used the tensor histograms of spatio-temporal images and the color histograms of frames as feature vectors. We selected three frames (the first, the middle, and the last frame) from one packet to extract the color features. The results are shown in Table 3. The accuracy after the 10th feedback for video 7 improved compared to the results in Table 1. This result shows the color features of frames are important for videos, especially, which have little motion such as video 7. Since video 8 has many types of scenes, shot changes, and little motion, the accuracy did not improve by adding the color features to temporal features. We should consider using other features to such types of videos.

4.4. Effectiveness of Improved Feedback Operations

We also examined the effectiveness of two feedback operations to reduce the user's effort to specify the relevance. By applying the feedback operation as shown in Fig.3, the accuracy was improved to 75% on average and the effort was decreased by about 90%. Next, we applied the other feedback operation as shown in Fig.4 setting n=5. As a result, the accuracy was improved to 80% on average and the effort was decreased by 80%. These experiments yielded promising results for our system to enable users to easily work with large video libraries.

5. CONCLUSION

We proposed a clustering method of video segments of a fixed length with relevance feedback to improve the clustering accuracy. Experiments with eight types of broadcasted videos showed that the accuracy was improved by about 24% on average after the 10th iteration. Although tensor histograms are effective for videos with lots of motion, color histograms need to be considered for other types of videos. We also conducted the experiments of applying the relevance feedback only to selected packets. As a result, the final accuracy was about 80% with 80% less effort. As a future work, we will consider how to select the representative packets which can more effectively improve the accuracy. Furthermore, we will also apply other feedback operations, for example, to combine similar two clusters.

6. REFERENCES

[1] H.Okamoto, Y.Yasugi, N.Babaguchi, and T.Kitahashi, "Video Clustering Using Spatio-Temporal Image with Fixed Length," IEEE International Conference on Multimedia and Expo, Vol.1, pp.53–56, Aug.2002.

[2] H.Lu and Y.P.Tan, "On Model-Based Clustering of Video Scenes Using Scenelets," IEEE International Conference on Multimedia and Expo, Vol.1, pp.301-304, Aug.2002

[3] C.W.Ngo, T.C.Pong, and H.J.Zhang, "On Clustering and Retrieval of Video Shots Through Temporal Slice Analysis," IEEE Transactions on Multimedia, Vol.4, No4, pp.446–458, Dec.2002.

[4] Y.Rui, T.S.Huang, M.Ortega, and S.Mehrotra, "Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval," IEEE Transactions on Circuits and Systems for Video Technology, Vol.8, No.5, pp.644–655, Sept.1998.

[5] N.D.Doulamis, A.D.Doulamis, and T.A.Varvarigou, "Adaptive Algorithms for Interactive Multimedia," IEEE Multimedia, Vol.10, No.4, pp.38-47, Oct.2003.

[6] Y.Lu, H.Zhang, L.Wenyin, and C.Hu, "Joint Semantics and Feature Based Image Retrieval Using Relevance Feedback," IEEE Transactions on Multimedia, Vol.5, No3, pp.339–347, Sept.2003.

[7] T.Kanungo, D.M.Mount, N.S.Netanyahu, C.D.Piatko, R.Silverman, and A.Y.Wu, "An Efficient k-Means Clustering Algorithm: Analysis and Implementation," IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.24, No.7, pp.881-892, July 2002.