To appear in: Proceeding of The 7th ACM International Multimedia Conference (Multimedia 99), Poster Session.
Orlando, Florida, USA. October 30-November 5, 1999.

A New Approach to Retrieve Video by Example Video Clip

Xiaoming Liu Yueting Zhuang Yunhe Pan

Institute of Artificial Intelligence, ZheJiang University
HangZhou, 310027
P. R. China
86-571-7951853

Liuxm@icad.zju.edu.cn Yzhuang@icad.zju.edu.cn Panyh@sun.zju.edu.cn

ABSTRACT

The similarity measure between video clips is a key issue in video retrieval. In the developing of our video retrieval system, we propose a new video similarity model. In contrast to existing algorithms, it proposes many influencing factors, such as order factor, speed factor, disturbance factor, etc, based on the subjective visual judgement of human. So this algorithm embodies the degree of similarity completely and systematically. On the other hand, it has resolution adaptation because it can be applied to every level of video structure. In the retrieval system, it can be used to process video query by example clip. This paper introduces it in detail and presents experiment results at the end of the paper.

Keywords

Video retrieval, example clip, similarity, factor.

1. INTRODUCTION

With the development of computer technique, the research of multimedia information retrieval has received attention from more and more researchers. In the image retrieval domain, many research fruits have been obtained from feature extraction to retrieval model[7]. As to the video retrieval, researchers usually focus on how to represent the content of video for its abundance and complexity[1]. But the work on the retrieval subsystem, the key to embody the efficiency and performance of system, has been overlooked relatively. To satisfy the user's query by example video clip, we propose a new video similarity model.

Currently the video retrieval usually follows three steps: 1) segment a video into a sequence of shot, and construct a hierarchical video structure, 2) extract the visual feature of keyframe and motion information, then store them into database, 3) process the user's query and return the results to the user. Let's focus on step 3. Young-II Choi et. al[6] had proposed a video data model, which integrated visual feature and annotated keyword, and a video retrieval language on the semantic level. But this language was not convenient to the user. Now the widely accepted query mode of video is keyword and example query. Example query is processed when a user submits a video or image and wants to find similar video. In the area of measuring the similarity between videos, Dimitrova et. al[3] regarded the average distance of corresponding frames between two videos as the similarity measure, and took the temporal order of the frames into account. Lienhart et. al[5] considered the video similarity from different hierarchies, and defined the measure by different degrees of aggregation based on either a set or sequence representation. But they did not consider other influences existing in the similarity measure.

In this paper we proposed a new model to measure the similarity between videos. As known from the research of psychology[4], the human's visual judgement has a lot of criterion. The similarity measure has significance only if it can simulate human's judgement. So our model comprises many sorts of influencing factors to embody the subjectivity of human. This paper is organized as follows. Section 2 introduces the construction of video database. The similarity model of video is detailed in section 3. Section 4 shows the experiment results. Finally we give the conclusions and future directions.

2. VIDEO DATABASE CONSTRUCTION

Video, a frame sequence on the temporal coordinate, contains abundant information. In our opinion, there are two sorts of feature, the local and global one, to represent the content of video. Firstly, the name, production date, director and some description texts of video belong to the global feature, which indicates the general attribution. On the other hand, the structure of video, i.e. the relationship between the shots, manifests the manner used in the production of video. This attribution of style also pertains to the global feature. Secondly, in the long temporal duration of video, every shot describes distinct content. So such content embodies the local feature of video, which is not only of the temporal, but also of the spatial.

The architecture of our video retrieval system is shown in figure 1. The storage subsystem is designed by the theme of global and local feature. At present, the general attributions of video, such as the name, production date, are annotated by human. We do shot-boundary detection by the integration of color histogram and edge extraction[9], and construct the structure of scene according to the semantic information. Within the shot, unsupervised clustering technique is used to do adaptive keyframe extraction[8]. After that, a video is represented as a hierarchical video structure consisted of the video, scene, shot and keyframe.

Based on the video organization, we extract visual features, color and texture, for every keyframe to construct the local feature database. As to the color feature, we first convert the color of every pixel from *RGB* space to *HSV* space, then calculate histogram using *H* and *S* weights in a 2D space, and finally regard 32 normalized values as the feature vector. We use the *coarseness*, *contrast* and *direction* defined by Tamura[2] to represent the texture feature, which is a vector consist of three values.

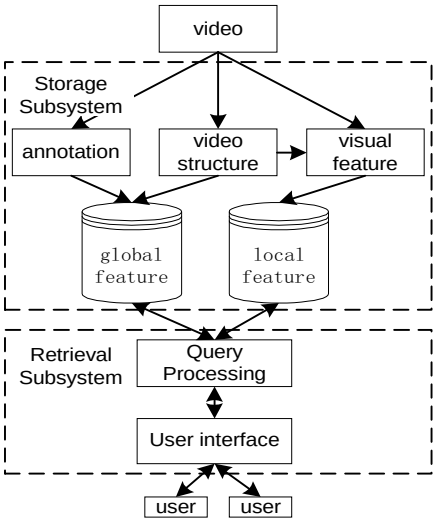

Figure 1. The architecture of video retrieval system

3. A VIDEO SIMILARITY MODEL

In the content-based retrieval, example query is a representative retrieval mode, in which a similarity model is needed. For the hierarchical structure of video, its similarity is measured by the lower layer, such as scene and shot. Here we do it based on the similarity of shots. As mentioned above, the judgement of visual similarity is a behavior related to human subjective and manifested many criterions. So let us see the criterions used in the human's subjective judgement of video similarity.

- Visual similarity. Two similar videos should be similar in the visual feature of low level, such as color, texture, etc. Usually it is the most important criterion. For example, in a video depicted the sun rising, the red keynote of color, and the circular shape of the sun are usually presented.
- Temporal order similarity. In two videos, some corresponding shots may have different temporal order though their contents are the same entirely. For example, the temporal order of three shots in $video_1$ is *ABC*, but in $video_2$ it is *CBA*, which means the first and last shot exchange their positions. Generally the human's preference about the temporal order will be manifested in the final similarity measure.
- Temporal duration similarity. Some films may have editions of different lengths after montage, within which there is the case that a shot has long temporal duration in the long edition while relative short one in the short edition. In fact it is the discrepancy of video content on speed. Generally when all other conditions are the same, the more alike the speeds are, the more similarity the videos have.
- Granularity similarity. Ideally when $video_1$ is similar to $video_2$, the same number of their shots has the correspondence of one-to-one. But the reality is not always the case. There will be a great diversity of the structure of correspondence, which influences the final similarity measure.

Suppose the video submitted by user is named a *query video*, which is denoted as V_q and segmented into n shots. So V_q is represented as:

$$V_q = \{S_{q1}, S_{q2}, \ldots, S_{qn}\} \quad (1)$$

where S_{qi} means the ith shot of V_q,. The temporal duration of S_{qi} and V_q is denoted as P_{qi} and T_q respectively. At the same time, there are h video clips in the video database. Anyone of them is named as a *database video* and denoted as V_d , which has m shots shown as:

$$V_d = \{S_{d1}, S_{d2}, \ldots, S_{dm}\} \quad (2)$$

The meaning of S_{di} and P_{di} is the same as before. And any shot, S_{di}, is represented by a keyframe, f_{di}, within itself. The objective of video retrieval is to find several video clips similar to V_q from h videos and return to the user in the descending order of similarity. This process follows four steps:

3.1 Construction of the corresponding graph

For every shot of V_q, S_{qi}, firstly we will find the most similar shot in V_d respectively. Because we represent a shot as a keyframe, the similarity measure between shots convert to the one between images. Now let us consider the latter. We convert the final similarity value in the range [0,1]. 1 means they have the most similarity, and 0 means they do not similar entirely. Because the attributions of two vectors composed the visual feature vector are different, we should distinguish them while calculating their distance.

As to the color vector, we can use histogram intersection as follow because it has been normalized in the range [0,1] and defined over the same physical domain.

$$Sim_{Color} = \sum_{i=0}^{31} Min(ColorHist_{f1}(i), ColorHist_{f2}(i)) \quad (3)$$

As to the vector-based feature representation, such as texture, every value does not in the same range. So we should convert all these values into the same range before the utilization of distance measure. By the example of *contrast* in texture feature, let us see how to use Gaussian normalization method. Suppose there are k keyframes, denoted as $Contrast_i (i=1\sim k)$, in video database. We first calculate the mean m and standard deviation σ of $Contrast_i$, then normalized k values into range [-1,1] by

$$Contrast'_i = \frac{Contrast_i - m}{3\sigma} \qquad i=0,..,k \quad (4)$$

As known from Gaussian distribution, the probability of $Contrast_i$ in the range [-1,1] is approximately 99%. So we should mapping the out-of-range values to either -1 or 1. Then we can use Euclidean distance to calculate the distance of the texture vector.

Now the respective distances of two vectors have been obtained. But only the color distance is in the range [0,1]. So we also have to normalize the texture distance by Guassian normalization. But after we use a formula like (4) to get $sim_{texture}$, the below liner transformation is needed for converting it to the range [0,1].

$$Sim'_{Texture} = 1 - \frac{Sim_{Texture} + 1}{2} \quad (5)$$

At last, we can get the similarity between two images according to two weights W_c, W_t as follow.

$$Similarity = W_c \bullet Sim_{Color} + W_t \bullet Sim'_{Texture} \quad (6)$$

Through it, every V_{qi} would find the most similar shot in V_d. We name every pair of similar shots as a correspondence. After this

section, for V_q and V_d, we can get a corresponding graph shown in figure 2. The connected solid line means this correspondence has the most similarity for V_{qi}.

3.2 Clustering-based similarity distinguish

After above calculation, we have obtained $n \times h$ similarity value, $Similarity_i$, between shots. But some of these values are very small, which means some shots of *query video* can not find similar shot in certain *database video*. If these small values are still included in the future calculation, the efficiency will be affected. So we want to distinguish a portion of values as similar values, others as no similar values, and remove the latter from the corresponding graph. But of these $n \times h$ values in the range [0,1] which pertains to the similar values? In fact, it equals to find a *cut* value in the range [0,1], and regard all the values as similar ones if they are larger than *cut*, other as no similar ones if smaller. Traditionally it is implemented by predefining a threshold. But here we use clustering-based algorithm to get the *cut* value. *Cut* is one value of the ascending order sequence of $n \times h$ $Similarity_i$, which can divide the sequence into two portions and let the centroids of two portions have the largest distance. The formula is shown as follow:

$$cut = \{Similarity_i \mid Max\left(\frac{\sum_{j=i+1}^{n \bullet h} Similarity_j}{n \bullet h - i} - \frac{\sum_{j=1}^{i} Similarity_j}{i}\right)\} \quad i=1,2,....,n \bullet h \tag{7}$$

Comparing to the method of predefining a threshold, this algorithm has a high flexibility. It can dynamically distinguish whether it is similar or not according to current situation of shot similarity. It is intuitionistic because similarity is a relative conception in nature. In figure 2, we use dashed line to represent shot correspondence of no similarity. In further calculation, their similarity values will be skimmed. The above two sections embody the visual similarity of human judgement.

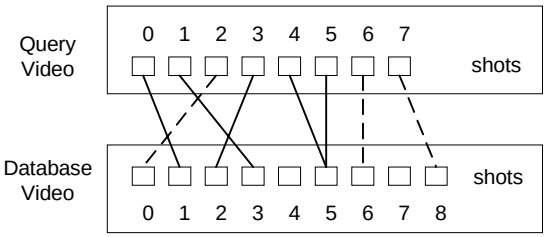

Figure 2. A corresponding graph of similar shot

3.3 Calculation of factors

Based on other criterions of human subjective judgement, we proposed several influencing factors to measure similarity.

The *order* factor is used to judge whether the corresponding shots of two videos appear in the same temporal order. As shown in figure 2, the $shot_3$ and $shot_2$ of *database video*, the corresponding shots of $shot_1$ and $shot_3$, are in a reverse order comparing to the *query video*. In the range [0,1], we use a value *order* to represent the degree of this reversion. Based on the corresponding graph, we get *order* by calculating the percentage of reverse correspondences to all the ones. 1 means the temporal order of shots appeared in two videos is the same. In the worst case, 0 means they are reverse entirely. This factor embodies the temporal order similarity of human judgement.

After the calculation in section 3.2, suppose there are *n'* and *m'* similar shots in V_q and V_d respectively. The *speed* factor is defined as follow:

$$Speed = 1 - \frac{|\sum_{i=1}^{n'} P_{qi} - \sum_{i=1}^{m'} P_{di}|}{\sum_{i=1}^{n'} P_{qi}} \tag{8}$$

In the range [0,1], this value describes the temporal duration descripancy of similar shots in two videos. 1 means their durations are the same. The closer to 0 the factor value is, the more descripancy their speeds have. It can be used to measure the descripancy degree of different film editions in speed. This factor embodies the temporal duration similarity of human judgement.

In the consecutive *n'* and *m'* corresponding shots, there are some shots do not have corresponding shot. For example, in figure 2, $shot_2$ of *query video* and $shot_4$ of *database video* pertain to this case, whose presentation indicates the discontinuousness of corresponding shots. So *disturbance* factor is utilized to measure the occurrence frequency of such shots. Suppose there are *x* and *y* such shots among *n'* and *m'* shots respectively. This factor is calculated as follow:

$$Disturbance = 1 - \frac{x+y}{n+m} \tag{9}$$

At last, we use the *congregate* factor shown in (10) to measure the frequency of one-to-many correspondence in the corresponding graph. For example, in figure 2, all the $shot_4$ and $shot_5$ of *query video* correspond to $shot_5$ of *database video*. This factor indicates the descripancy of depiction degree of similar shot in different videos. The last two factors manifest the granularity similarity of human judgement.

$$Congregate = \frac{m'}{n'} \tag{10}$$

3.4 Calculation of final similarity

Based on the above analysis, we can calculate the similarity between V_q and V_d as follow:

$$Similarity = W_1 \bullet \sum_{i=1}^{n'} \left(\frac{P_{qi}}{T_q} \bullet Similarity_i\right) + W_2 \bullet Order + W_3 \bullet Speed + W_4 \bullet Disturbance + W_5 \bullet Congregate \tag{11}$$

where $W_1,...,W_5$ indicate the preference of the user regarding to every judgement criterion. So the initial weights can be determined not only by the system, but also by the user through indicating his preference about those four criterions on the user interface. The fraction before $Similarity_i$ is the percentage of $shot_i$ duration to the video duration. This weight manifests that long duration shot will have more contribution to the final video similarity, which accords with the mental judgement of human. When the calculation of similarity between V_q and any video V_d in database is finished, we can return several most similar V_d to the user in the descensive order of similarity. Thus the query processing is finished.

From it we can see that our algorithm can be applied to any layer of the video structure although we show its utilization in the calculation of video similarity, i.e. it is generally and has a resolution adaptation. For example, we can use this algorithm to calculate shot similarity based on the frame sequence. On the

other hand, if a user submits an image as an example query, this algorithm can still work well because an image can be regarded as a special form of video.

4. EXPERIMENT RESULTS

Recently, We have implemented a video retrieval system, *vilib*, on personal computer. As shown in figure 1, this system is divided into two parts. Based on the SQL Server database, the storage subsystem is implemented with visual C++, and the retrieval subsystem is done with Active Server Pages.

Our video database consists of 5 hours of video from various sources. Within it, there are films such as "Forest Gump" and "True Lies", the series, several TV commercials, and live recordings. For each video clip, we calculate its features and storage them to the database.

We test our system by submitting several video clips as example query. One of the query results is shown in figure 3. In every row, we use keyframe to represent the video clip. The clip in the first row is submitted by the user, and others are the similar clips found by our system. As we can see, in such a big video testbed, all these results are relevent to the *query video* and they listed in a descensive order of similarity. Based on the human's relevance judgement, we get the *recall* and *precision* value for four example queries, and show them in table 1. From it we can see the excellent performance of our system. We also observe that our algorithm is especially appropriate for small clip retrieval.

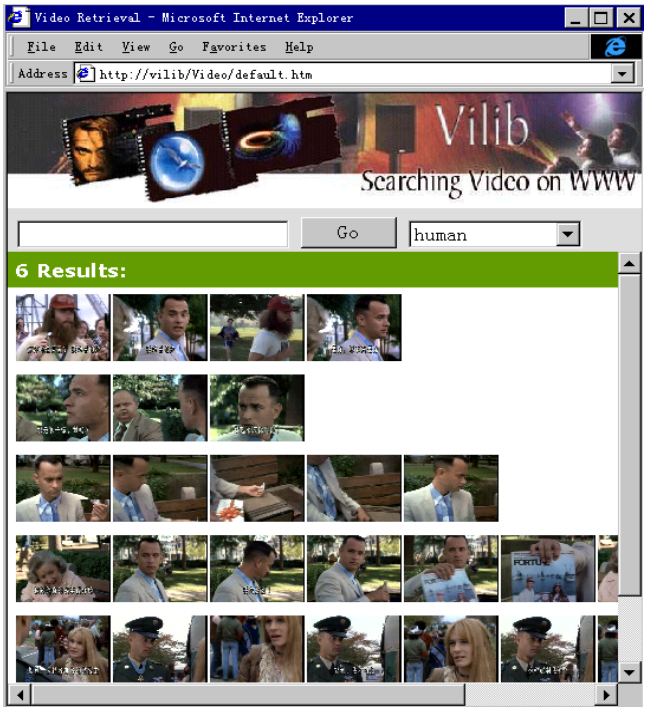

Figure 3. A retrieval result of our system

Query	The content of query video	Recall	Precision
1	Gump talks with someone	0.9	0.84
2	Someone is playing Ping-Pong	0.7	0.6
3	Sun is rising	0.7	0.65
4	A man is walking on the meadow	0.8	0.6

Table 1. Retrieval performance of four example queries

5. CONCLUSIONS

The similarity measure of video clip is a key issue in video retrieval. In this paper we presented a new similarity model. Comparing to existing algorithm, it proposes many influencing factors, such as order factor, speed factor, etc, based on the human's subjective visual judgement. So this algorithm embodies the similarity degree completely and systematically. On the other hand, it has resolution adaptation because it can be applied to every level of video structure. In the retrieval system, it can be used to process video query by example clip successfully, which has been proved in our system. Now, we are doing further work on the relevance feedback, through which the weights of every factor can be adjusted according to the user's feedback. In such a mechanism, this algorithm can embody the user's preference more accurately and return better retrieval results to the user.

6. ACKNOWLEDGEMENTS

Our work was sponsored by the National Natural Science Foundation of China. We would also like to thank Yi Wu and Yi Mao for fruitful discussions.

7. REFERENCES

[1]. Hong Jiang Zhang, JianHua Wu, Di Zhong and Stephen W.Smoliar, An Integrated System for Content Based Video Retrieval And Browsing, Pattern Recognition, Vol.30, No.4, 1997, 643-658.

[2]. H.Tamura, S.Mori, T.Yamawaki, Texture features corresponding to visual perception, IEEE Trans. on Sys, Man, and Cyb, Vol.SMC-8, No.6,1978.

[3]. Nevenka Dimitrova and Mohamed Abdel-Mottaled, Content based Video retrieval by example video clip, In:SPIE Vol. 3022,1998.

[4]. S. Santini and R.Jain. Similarity Matching. submitted to : IEEE trans. on Pat. Ana. and Mac. Int. URL: ftp:// vision.ucsd.edu/pub/simone/vision/SimPaper.ps.Z.

[5]. R. Lienhart, W. Effelsberg and R. Jain, VisualGREP: A systematic method to compare and retrieval video sequences, In: SPIE Vol.3312, 1997, 271-282.

[6]. Young-II Choi, Yoo-Mi Park, Hun-Soon Lee and Seong-II Jin, An Integrated Data Model and A Query Language for Content-Based Retrieval of Video, In: Proceedings of MIS'98, Istanbul, Turkey, Sept. 1998, 192-198.

[7]. Yong Rui, Thomas S. Huang and Shih-Fu Chang, Image Retrieval: Current Techniques, Promising Directions and Open Issues, Journal of Visual Communication and Image Representation, 10,1999.

[8]. Yueting Zhuang, Yong Rui and Thomas S. Huang, Adaptive Key Frame Extraction Using Unsupervised Clustering, IEEE ICIP'98, Oct.1998, Chicago, USA.

[9]. Zhuang Yueting, Wu yi and Pan Yunhe, Video Catalog--A New Approach to Video Organization, accepted for publication in Pattern Recognition and Artifical Intelligence, Dec. 1999.