

A Novel Approach for Query by Video Clip

Deepak C R¹, Sreehari S², Sambhu S Mohan³

^{1,3}(Computer Science, Amrita school of engineering/ Amrita vishwa vidyapeetham, India)

²(Electrical and Electronics, Amrita school of engineering / Amrita vishwa vidyapeetham, India)

Abstract : In this paper we propose an efficient algorithm to retrieve videos from the database when a video clip is given as query. A retrieval system should have high precision, recall and low search time complexity. In this paper search time complexity is reduced by using clustering and search refinement method. Experimental results show that proposed video retrieval method is efficient and effective. Spatial and temporal properties of the video are used to retrieve the videos from the database.

Keywords - video indexing, Video retrieval

I. INTRODUCTION

One of the main challenges in query by video clip is the time complexity in searching due to the huge amount of data that are present in the database. In order to make the retrieval system efficient we need algorithms which have low time complexity in searching. Most of the algorithms uses only properties of the key frames and doesn't use the temporal properties of the shots; this reduces the precision of the retrieval system. Sliding window algorithm is the one of the initial method that is used for searching [1], but it is time consuming process.

For efficient video indexing and retrieval, video shot boundary detection is the main step, which divides video into small video blocks which conveys same visual information. Keyframe selection from the video shots reduces the redundancy of features in feature database. Temporal features that we have used include motion changes within a shot, shot length, and sequence of shots in the video. There are many content based video retrieval systems which are based on motion [2], [3], [4] and are efficient in retrieving videos. Here we have used various spatio-temporal features to increase the accuracy of the retrieval system.

II. PROPOSED METHOD

Efficiency of content based video retrieval system is based on features extracted, indexing, and searching methods used to retrieve the videos. In order to get high precision and to reduce searching time we have used clustering and search refinement method. Feature selected in this project gives best representation of the video shot. Overall architecture of the system is given in Figure 1.

1.1 Temporal video segmentation

There are many efficient algorithms to divide videos into shots [5], [6], [7], [8], [9]. Videos in the database are temporally segmented using block based histogram. This is done to get smallest video block which is easy to index. Since visual information in all the frames in a shot is almost same, one frame is enough to represent visual information in the temporally segmented block. Middle frame in a shot is selected as keyframe for that shot. Frame number of the keyframe is found out using the following equation.

$$\text{FrameNumber(KeyFrame)} = (\text{FrameNumber(BlockEnd)} - \text{FrameNumber(BlockStart)}) / 2$$

1.2 Feature Extraction

Color, texture edge and motion features are extracted from the key frames. For color, block based color histogram and color moments (color mean, color standard deviation and color skewness) are used. Before extracting RGB color histogram, keyframe is quantized into 78 levels. Block based histogram is used in order to get spatial distribution of color. Global histogram based method does not give information about spatial distribution of color.

Texture properties are extracted using Gray Level Co-occurrence Matrix (GLCM). The texture features that are extracted from the GLCM matrix are angular second moment, maximum probability, moments, inverse moment, contrast, homogeneity, entropy and correlation.

Edge information gives the information about the orientation of the objects present in the keyframe. Canny edge detector is used to get the edge information. After applying canny edge detection algorithm, keyframe is divided into small blocks and edge pixels in each block is calculated, division of frames into small blocks gives the positional information of edges present in frames.

Motion information in a video is taken from keyframes and also from the entire shot. Motion vectors are extracted from the keyframes using Lucas and Kanade's algorithm. Lucas and Kanade's optical flow

algorithm has good performance compared to others [10]. Motion vectors in the key frame extracted using Lucas and Kanade’s optical flow algorithm are given in figure 2.

Motion in a shot is classified in to smooth, small, fast and irregular. A single shot may contain different types of motions. Sometimes in the beginning of the shot motion may be smooth but may be at the end motion may be irregular, so motion in a shot is represented by a string. Suppose a shot have three types of motions smooth motion in the beginning, irregular motion in the middle and small motion at the end. These motions represented by string as {irregular, smooth, small}. Length of the shot is calculated by extracting number frames in a shot and frame per second in a video. Suppose there is N number of consecutive frames which having same visual information, and if number of frames per second is T, then the shot length in seconds can be calculated using following formula.

$$\text{Shot length} = N/T$$

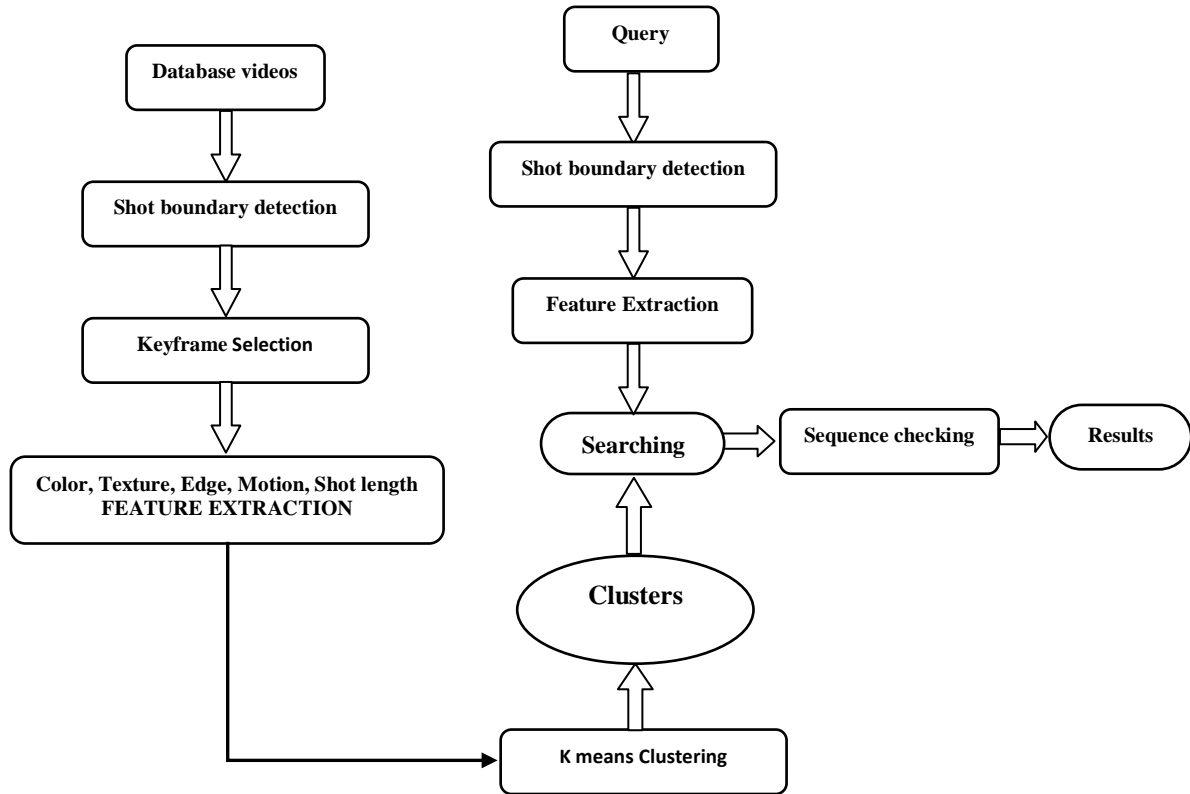


Fig 1: Overall system architecture

One of the main temporal features of the video is sequence of shots that are appearing in video. A checking on these sequences is performed while retrieving the video from the database. Suppose A, E, B, D are the sequence of shots in query video clip. Then the retrieved videos should also follow these sequences.

Table 1: Dataset details



Fig 2: Motion vectors in frame

Movie Name	Time (HH:MM:SS)	Frames	Shots
Usthad	2:12:16	198400	3517
MI 3	1:02:03	85629	1868
Daisy	2:04:34	186850	3125
Thirakatha	2:07:56	176548	3392

Query



RESULTS

Rank - 1



Rank - 2



Rank - 3



Fig 3 : Results

Shot length feature gives the length of the each shot. Shot length of first and last shot in a query is not extracted for matching shot length between shots in database and query. Only the shots which lie between first and last shot are considered for extracting shot length. Suppose our query is part of movie, in this case first and last shot of the query may contain only a part of original shot.

1.3 Clustering and searching

Searching through all the shots in a video one by one is time consuming due to the dimensionality of the feature vectors and huge amount of shots in video. A normal two hour long movie contains more than one lakh frames and more than one thousand of shots. So searching in videos using sliding window manner is time consuming and impossible to implement in real time systems. Here we are applying a clustering and hierarchical searching method to reduce the time complexity in retrieving.

After extracting features K means clustering algorithm is applied. Clustering in shots is done based on the color histogram (fig 4). Initial centroids and number of clusters are selected manually. For making clusters using K-means algorithm, the distance of each shot to the every cluster centroid is calculated. Each shot is put in to the cluster which having smallest distance. Centroid of the cluster is updated after every new entry to the cluster. Euclidean distance is used to measure the distance between histogram of two keyframes.

Color histogram of the query shot is found and the cluster which is closest to the color histogram of query video shots is selected for further processing. And other clusters are eliminated from the further searching process. Texture information in the keyframes of the selected clusters is compared with query key frames. If the similarity distance from the query keyframes is more than particular threshold T_1 , then that shots are discarded for next level of searching. Likewise lower dimensional features are compared first and at each level unmatched

shots are discarded. Using this method matching between shots using high dimensional feature vectors is done only for fewer numbers of shots. So time complexity in similarity matching of high dimensional feature vectors is considerably reduced. This reduces overall search time complexity.



Fig 4: Clusters of video shots

III. EXPERIMENTAL RESULTS

The video database of our experiments comprises as shown in Table 1, which includes three Malayalam movies and one English movie. There are a total of 11902 shots, and searching is done using 35 video clips. Euclidean distance is used to measure the distance between feature vectors for each query precision and recall is calculated using following formula.

$$Precision = \frac{\text{No. of relevant videos retrieved}}{\text{No. of retrieved videos}}$$

Mean precision percentage calculated from 35 query result is found to be 73.65.

IV. CONCLUSION

In this paper we present a novel content based video retrieval system using the conventional video features such as color, texture, edge and motion patterns. In this content based video retrieval system query is a video clip. Our system implemented over a large video database produces satisfactory results due to spatio-temporal features along with clustering algorithm.

REFERENCES

- [1] L. P. Chen, T. S. Chua, "A match and tiling approach to content-based video retrieval.: Proc. ICME 2001, pp. 301-304, Aug. 2001
- [2] M. Flickner *et al.*, "Query by image and video content: The QBIC system," *IEEE Comput. Mag.*, vol. 28, pp. 23-32, Sep. 1995.
- [3] S. F. Chang, W. Chen, H. J. Meng, H. Sundaram, and D. Zhong, "A fully automated content-based video search engine supporting spatiotemporal queries," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 5, pp. 602-615, Sep. 1998.
- [4] S. Dagtas, W. Al-Khatib, A. Ghafoor, and R. L. Kashyap, "Models for motion-based video indexing and retrieval," *IEEE Trans. Image Process.*, vol. 9, no. 1, pp. 88-101, 2000.
- [5] C.-W. Su, H.-Y. M. Liao, H.-R. Tyan, K.-C. Fan, and L.-H. Chen, "A motion-tolerant dissolve detection algorithm," *IEEE Trans. Multimedia*
- [6] C.-C. Shih, H.-R. Tyan, and H.-Y. M. Liao, "Shot change detection based on the Reynolds transport theorem," *Lecture Notes in computer Science*, vol. 2195, pp. 819-824.
- [7] Bing Han, Xinbo Gao, Hongbing Ji, 2005. A shot boundary detection method for news video based on ough-fuzzy sets. *Int. J. Inform. Technol.*, 11: 101-111. www.icis.ntu.edu.sg/scs-jjit/117/117_11.pdf
- [8] Gao, X. and X. Tang, 2002. Unsupervised videoshot segmentation and model-free anchorperson detection for news video story parsing. *IEEE Trans. Circuits Syst. Video Technol.*, 12: 765-776. Doi: 10.1109/TCSVT.2002.800510
- [9] Gao, X. and X. Tang, 2000. Automatic parsing of news video based on cluster analysis. In: *Proceedings of 2000 Asia Pacific Conference on Multimedia Technology and Applications*, Kaohsiung, Taiwan, China, Dec. 17-19, pp: 17-19. <https://dspace.lib.cuhk.edu.hk/handle/2006/5923>
- [10] J. L. Barron, D. J. Filt, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43-77, 1994.