Content Based Video Retrieval Using Cluster Overlapping

Deepak C R[1], Sreehari S[2], Gokul M[3], Anuvind B[4]
1, 3(Computer Science, Amrita School of Engineering/ Amrita vishwa vidyapeetham, Coimbatore, India)
2(Electrical and Electronics, Amrita School of Engineering/ Amrita vishwa vidyapeetham, Coimbatore, India)
4(Electronics and Communication, Amrita School of Engineering/ Amrita vishwa vidyapeetham, Coimbatore, India)

ABSTRACT:

To retrieve videos from database efficient video indexing and retrieval mechanisms are required. In this paper, we propose an efficient algorithm to retrieve videos from the database when a video clip is given as a query. To efficiently match query video clip with the videos in the database various spatio-temporal features are used. Clustering algorithms are applied to extracted features for fast retrieval. Cluster overlapping method is used to retrieve relevant videos. Relevant videos are ranked based on the similarity measurement and frequency of query shots in the retrieved videos. Experimental result proves that proposed method has high precision and recall compared to conventional algorithms.

Keywords: Content based video retrieval, key frame, video indexing, shot clustering

I. INTRODUCTION

There are various content based video retrieval and indexing methods which uses spatio-temporal features. For video indexing, Jianping Fan et al. [1] proposed hierarchical video shot classification method. In hierarchical video shot classification method video is structured in to hierarchical tree using semantics of the video. H. Farouk et al. [2] proposed video indexing and retrieval algorithm using wavelet coefficients that are extracted from video frames. Ja-Hwung Su et al. [3] proposed video indexing and retrieval mechanism using pattern trees constructed from the video shot patterns in a video. They have used fast pattern index tree and advanced pattern index tree for indexing and ranking of the videos. Stacie Hibino and Eke A. Rundensteiner analyzed videos in terms of temporal relationship between events [4]. First step in video indexing and retrieval is temporal video segmentation. Various methods are proposed for temporal video segmentation [5], [6], [7], [8], which are efficient in dividing video into shots.

II. PROPOSED METHOD

In the proposed system we have used various spatio-temporal features of the video and clustering methods to efficiently retrieve videos from the database when a video clip is given as a query. Intersection of various clusters is found in order to find the relevant videos from the database. The shots which are present in the intersection of relevant clusters are ranked based on the similarity between query shots and their frequency of appearance in the videos. Detailed architectural diagram is given in figure 1.

2.1. Spatio-temporal features

Database videos are divided into shots using combined color, edge and motion features [9]. Using these features, adjacent frames in video are compared in order to find the shot boundary. One key frame is selected from the each shot using mutual information and image entropy [10]. Key frames selected from videos are given in figure 2. For getting color information from keyframes, RGB color histogram is used. Before the extraction of color histogram each frame in a video is quantized in to 86 intensity levels. Texture information from a frame is calculated using tamura features [11]. Features that are extracted for texture information are coarseness, directionality, regularity and degree of contrast. Canny edge detector is used to extract edge information [12]. For extracting edge features each frame is divided into 16 sub-blocks and edge direction and its count from each block is calculated. Edge direction information is obtained by convolving edge image with various filters. Optical flow algorithm is used for getting motion vectors from the keyframe. Another temporal feature extracted from video shots is camera motion. Camera motion in a video is classified into irregular camera motion, still camera, smooth camera, little camera motion and no camera motion [13]. In the same manner motion vectors in the key frame is classified into four type no motion, small motion, medium motion and large motion. After finding motion vectors using Lucas Kanade's algorithm frame is divided into 16 sub-blocks. For each block based on the motion vectors we determine motion of that block as one among the four types of motion vectors- no motion, large motion, small motion, and medium motion.

2.2. Clustering

After feature extraction shots are clustered, using features extracted. Let S_1, S_2, S_3,......,S_N be the shots in the database after temporal segmentation of videos in the database. Using RGB color histogram feature these shots are grouped in to M number of clusters. K-means clustering algorithm is used for clustering the shots. The same N number of temporally segmented shots is again clustered using texture properties. This clustering process is also done using motion pattern in keyframe, camera motion and edge features.

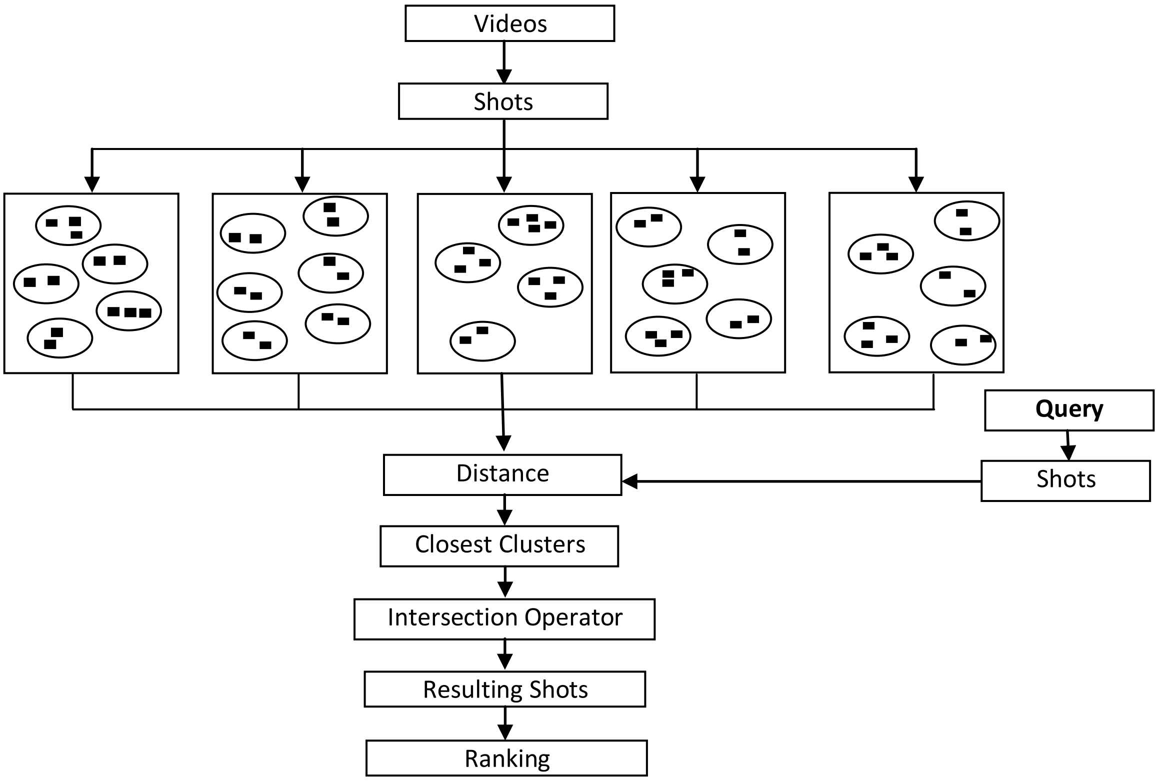

Fig. 1. Overall system architecture

Now we have 5 clusters video shots using the features RGB color histogram, texture, camera motion, motion vectors from keyframe and using edge information. Let CH_1, CH_2, CH_3, ..., CH_p be the clusters using color histogram, T_1, T_2, T_3,.....T_q be the clusters using texture, E_1, E_2, E_3,....E_r be the clusters using edge information, MP1, MP_2, MP_3,.........., MP_t be the clusters formed using motion pattern and MC_1, MC_2, MC_3,.......MC_v are the clusters formed using camera motion pattern.

Let Cs={ CH_1, CH_2, CH_3, ..., CH_p }, Ts={ T_1, T_2, T_3, ..., T_q }, Es={ E_1, E_2, E_3, ..., E_r}, Ms={ MP1, MP_2, MP_3, ..., MP_t } and Ps={ MC_1, MC_2, MC_3, ..., MC_v}. After extracting features from the query shots, one closest cluster from Cs, Ts, Es, Ms and Ps is selected for each query shot. Let CH_a, T_b, E_c, MP_d and MC_e are the clusters which is close to the first shot in the query, where $1< a <p$, $1< b <q$, $1< c <r$, $1< d <t$ and $1< e <v$.

Let S be the set of shots which is contained in CH_a, T_b, E_c, MP_d and MC_e. The set S is defined as intersection of CH_a, T_b, E_c, MP_d and MC_e i.e., For first shot in the query, the shot result set is given by $S = CH_a \cap T_b \cap E_c \cap MP_d \cap MC_e$. Let S= {$S_1$, S_2, S_3,.....,S_{lm}} be the shot result set for first shot in the query where S_i, i $\in$ [1..lm] represents the shots which are similar to the first shot in the query. If $|S| < R_{Thres}$ or $S = \varnothing$ then the set is updated to $S = S \cup (CH_a \cap E_c \cap MP_d \cap MC_e)$. If still $|S| < R_{Thres}$ or $S = \varnothing$ then S is updated to $S = S \cup (CH_a \cap E_c \cap MP_d)$. If still $|S| < R_{Thres}$ or $S = \varnothing$ then again S is updated to $S = S \cup (CH_a \cap E_c \cap MP_d)$. Where R_{Thres} is the minimum number of results required in the result set. The similarity between shots is measured using Euclidean distance.

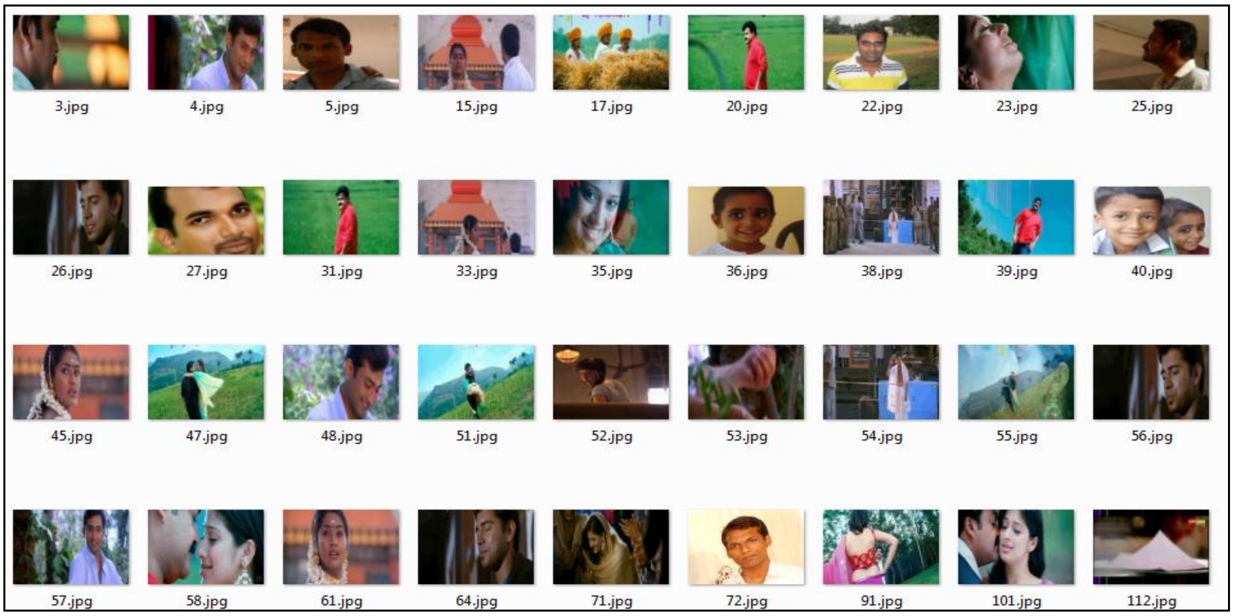

Fig. 2: Keyframes

Let database videos be denoted as V={$Video_1$, $Video_2$, ...,$Video_z$}, and each video contains a set of shots. A video clip $Video_i$ is represented as$Video_i$= {$shot_k$, $shot_{k+1}$, ...,$shot_j$}, where $Video_i \epsilon V$. Let us assume that there are four videos in the database and each video contains sets of shots as in Table 1.

Table 1

Video	**Shots**
$Video_1$	$shot_1$,$shot_2$,$shot_3$,$shot_4$,$shot_5$
$Video_2$	$shot_6$,$shot_7$,$shot_8$,$shot_9$,$shot_{10}$
$Video_3$	$shot_{11}$,$shot_{12}$,$shot_{13}$,$shot_{14}$,$shot_{15}$
$Video_4$	$Shot_{16}$,$shot_{17}$,$shot_{18}$,$shot_{19}$,$shot_{20}$

For example, assume that query video clip contains 4 shots, denoted as follows {qShot1, qShot2, qShot3, qShot4}.For the first shot in the query, qShot1, the retrieved shots and their similarity measured using above method is given in Table 2.

Table 2

Query shot	Retrieved Shot	Video	Similarity%	Appear_val
qShot1	$Shot_7$	$Video_2$	90	1
	$Shot_2$	$Video_1$	88	1
	$Shot_{16}$	$Video_4$	76	1
	$Shot_{20}$	$Video_4$	57	2
	$Shot_5$	$Video_1$	45	2

Table 3, Table 4, Table 5 shows retrieved shots for the second, third and fourth shot in the query respectively.

Table 3

Query shot	Retrieved Shot	Video	Similarity%	Appear_val
qShot2	$Shot_8$	$Video_2$	93	1
	$Shot_3$	$Video_1$	82	1
	$Shot_{17}$	$Video_4$	72	1
	$Shot_{13}$	$Video_3$	56	1
	$Shot_2$	$Video_1$	32	2

Table 4

Query shot	Retrieved Shot	Video	Similarity%	Appear_val
qShot3	$Shot_9$	$Video_2$	91	1
	$Shot_4$	$Video_1$	73	1
	$Shot_{18}$	$Video_4$	59	1
	$Shot_3$	$Video_1$	52	2
	$Shot_{17}$	$Video_4$	49	2

Table 5				
Query shot	Retrieved Shot	Video	Similarity%	Appear_val
qShot4	$Shot_5$	$Video_1$	88	1
	$Shot_{10}$	$Video_2$	84	1
	$Shot_{19}$	$Video_4$	76	1
	$Shot_{18}$	$Video_4$	72	2
	$Shot_6$	$Video_2$	57	2

	Table 6	
Video	Similarity	Rank
$Video_1$	369.83	2
$Video_2$	375.15	1
$Video_3$	56	4 4
$Video_4$	321.83	3

The videos are ranked based on the visual similarity and number of times shots are appeared in the retrieved shots.Rank of each video is calculated by summing highest rank of video in each table. The final rank of each video is calculated by using following formula.

$Fin\_Rank(Video_i)$= ($\sum$Highest similarity from each table corresponding to $Video_i$) $+\sum$ (similarity$\times$ log(Appear_val)).

$Video_1$=88+82+73+88+(88)(log1)+(45)(log2)+(82)(log1)+(32)(log2)+(73)(log1)+(52)(log2)+(88)(log1)=369.83 , $Video_2$=90+93+91+84+(57)(log2)=375.15, $Video_3$=56, $Video_4$=76+72+59+76+(57)(log2)+(72)(log2)=321.83.

Query

Rank-1

Rank-2

Rank-3

Rank-4

Fig. 3 : Result

	Table 7	
Video	No. of frames	No. of shots
Movie 1	41548	1598
Movie 2	49500	1980
Movie 3	42840	1785
Football	28896	1204
Cricket	23472	978
Campus video	27700	1108

III. EXPERIMENTAL RESULTS

The video database of our experiments is summarized in Table7. The number of clusters for each feature is selected manually. In the experiments performed, the average precision of 85% and 76% recall were obtained. Figure3shows frames from a video query clip and retrieved video clips. The retrieved video clips are very close to that of a query clip. The experimental results are shown in figure 4.

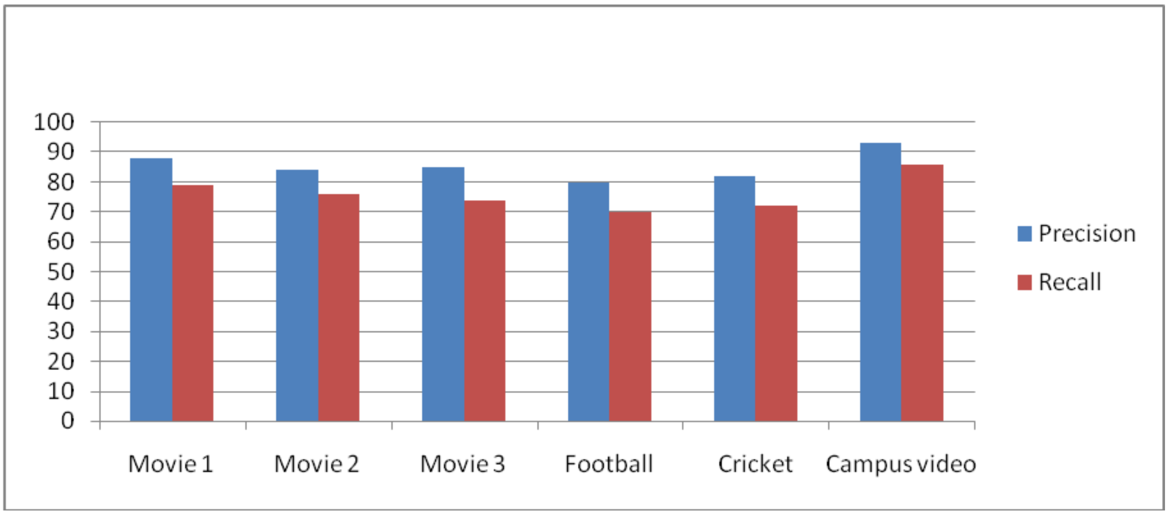

Fig. 4. Performance

REFERENCES

[1] Jianping Fan, Ahmed K. Elmagarmid,Xingquan Zhu, Walid G. Arefand Lide Wu *"Class View*: Hierarchical Video Shot Classification, Indexing, and Accessing," *IEEE TRANSACTIONS ON MULTIMEDIA*, VOL. 6, NO. 1, FEBRUARY 2004.

[2] Farouk, H.; Elsalamony, H. A., "Digital library creation based on wavelet coefficients for video stream indexing and retrieving," *Signal Processing Systems (ICSPS), 2010 2nd International Conference on* , vol.1, no., pp.V1-158,V1-162, 5-7 July 2010

[3] Ja-Hwung Su, Yu-Ting Huang, Hsin-HoYeh, Vincent S. Tseng "Effective content-based video retrieval using pattern-indexing and matching techniques,"*Expert Systems with Applications* 37 (2010) 5068–5085.

[4] Hibino, S.; Rundensteiner, E.A., "A visual query language for identifying temporal trends in video data," *Multi-Media Database Management Systems, 1995. Proceedings., International Workshop on* , vol., no., pp.74,81, 28-30 Aug 1995.

[5] ZhiyiQu; Ying Liu; LipingRen; Yong Chen; RuidongZheng, "A method of shot detection based on color and edge features," *Web Society, 2009. SWS '09. 1st IEEE Symposium on* , vol., no., pp.1,4, 23-24 Aug. 2009.

[6] *Junaid Baber, NitinAfzulpurkar, Matthew N. Dailey, and MaheenBakhtyar*, "Shot boundary detection from videos using entropy and local descriptor," *Digital Signal Processing (DSP), 2011 17th International Conference on* , vol., no., pp.1-6, 6-8 July 2011.

[7] G.G.LakshmiPriya, S.Domnic, "Video Cut Detection using Hilbert Transform and GLCM," *IEEE-International Conference on Recent Trends in Information Technology*, ICRTIT 2011 978-1-4577-0590-8/11.

[8] WenzhuXu, LihongXu, "A novel shot detection algorithm based on clustering," *Education Technology and Computer (ICETC)*, 2010 2nd International Conference on , vol.1, no., pp.V1-570-V1-572, 22-24 June 2010.

[9] Jacobs, A., Miene, A., Ioannidis, G. T., and Herzog, O. Automatic shot boundary detection combining color, edge, and motion features of adjacent frames. In TRECVID 2004, pp. 197--206, 2004.

[10] Lina Sun; Yihua Zhou, "A key frame extraction method based on mutual information and image entropy," *Multimedia Technology (ICMT), 2011 International Conference on* , vol., no., pp.35,38, 26-28 July 2011

[11] Tamura, Hideyuki; Mori, Shunji; Yamawaki, Takashi, "Textural Features Corresponding to Visual Perception," *Systems, Man and Cybernetics, IEEE Transactions on* , vol.8, no.6, pp.460,473, June 1978

[12] Canny, John, "A Computational Approach to Edge Detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* , vol.PAMI-8, no.6, pp.679,698, Nov. 1986

[13] Haoran Yi , DeepuRajan , Liang-Tien Chia, A new motion histogram to index motion content in video segments, Pattern Recognition Letters, v.26 n.9, p.1221-1231, 1 July 2005