MOTION TRAJCECTORY BASED VIDEO CONTENT RETRIEVAL AND DELIVERY FOR SMALL DISPLAYS

P. Geetha, Dr. Vasumathi Narayanan

ABSTRACT

Adaptive Multimedia Content Retrieval and Delivery for small displays is one of the challenges faced by Multimedia Community. Input video is transformed to an output video by utilizing manipulations at multiple levels (signals, structural or semantics) to meet diverse resource constraints and user preferences with optimizing overall utility of the video. The proposed system is developed to display the retrieved video shot, by motion trajectories of individual object, in a small displays. This system needs video shots as the inputs whose motion vectors are extracted by using exhaustive search algorithm. This shot-level motion feature is linked across the consecutive frames of shot to form the motion trajectories. Remove redundant trajectories and preserve one motion trajectory from all the similar motion trajectories. The representative object motion trajectory is stored in a database. Query interface which allows users to search for similar video shots by giving query video clip as input. Similarity matching algorithm is used to retrieve similar video shot from the database by comparing their motion trajectories. In this paper, next, in order to display those retrieved video shots in a small display, shape information of moving objects are extracted using Region- Growing algorithm. The segmented foreground is scaled down and re-integrated with the repaired and directly resized background to deliver effective video shot for small displays.

Keywords: Motion Trajectory, Exhaustive Search algorithm, Douglas - Peucker algorithm, Region Growing algorithm, Content-based video retrieval (CBVR).

1. INTRODUCTION

The advent of multimedia computing has led to increased demands for digital videos. CBVR plays a crucial role in many multimedia / data mining applications and is an important but inherently difficult problem. Video Adaptation is a key technology for transforming video stream with one or preferences. At user's end, hand-held devices including cellular phones, PDAs, and pocket PCs are now used for their mobility and portability. In order to display in these hand-held devices, the screen size is kept permanently unchanged and even as small as possible. For adapting videos on a small display a conventional scheme can be divided into two categories: spatial transcoding and frame cropping [1][2]. Due to bias of their design purpose, an adaptation engine has to make visual tradeoffs between the subject readability and content completeness [3]-[5]. The difficulties in conventional scheme arise because they both passively attempt to adopt the plain frame but not the actual content it contains. A novel framework is proposed by Wen-Huang Cheng et al. to provide effective small videos which emphasize the important aspects of a scene with retaining the background context. Adaptive content delivery for small displays aims at delivering effective small size videos for its displays. Video segmentation is done by extracting the shape information of moving object from the video sequence. A foreground object is defined as a region and other areas are regarded as uncovered background regions and are discarded. The thresholds of distance measure determine how homogeneous region should be at the end of that frame. Small thresholds tend to generate multiple colors consistent but small regions. Such thresholds cause over-segmentation. On the other hand, larger thresholds may combine colored regions into one by passing over weak edges, which bring in under-segmentation. Distance threshold controls the color variance of the region.

Interpolation is an imaging method to increase or decrease the number of pixels in a digital image. Many method are available to rescale foreground frames of video ie., Bilinear Interpolation, Bicubic Interpolation etc., Nearest neighbors interpolation is the method is used in [3], which finds the location of each new point relative to one nearest neighbor of the original image and makes the pixels bigger. It does not change the color information of the image and does not introduce any anti-aliasing.

The background is assumed to be stationary and the filling of stationary background is done by computing the confidence of pixels which are deemed to belong to the moving foreground or to the damaged area is set to zero. The rest of the pixels are initialized to a confidence value of one. The static background filled-in is consistent throughout the video [2][3].

This adaptive video to display on the small display is retrieved based on their content. There are many retrieval techniques exist in survey[6]. Mainly Multimedia databases are convenient for describing and storing media such as image, text, voice, video temporal and physical components of information. The large amount of multimedia information requires efficient and effective annotation and retrieval method. There are several applications that need the ability to query these multimedia objects based on their content. Visual content based retrieval is best utilized when compared with the traditional search [7], at user interface and the system level. The basic reason for this is that we do not see the possibility of content based retrieval replacing the ability of parametric search or text search. The key is to apply content based retrieval wherever appropriate, and this is where the use of text is become suboptimal. In Multimedia applications, visual appearance (e.g. color, texture, shape, motion) is an important search argument. The determination of features, which represent the issues, is still a serious problem, which affects the results of retrieval directly. There are many types of feature extraction methods in literature. These features are extracted from video by many ways. However, there is not any unique method that fits all issue types. Popular feature extraction methods include auto-correlation [8], gray-level based approaches [9, 10], co-occurrence matrices [11], wavelet-based transform [12], Discrete Fourier Transform [13], and discrete cosine [14] transforms. Main problems of video processing applications come from the large number of features with position and scaling changes. Therefore, more clever and robust feature extraction methods are needed.

As motion is an important feature of video, much work has been carried out on motion feature video retrieval [15] [16]. This motion feature is linked across the consecutive frames of shot to form the motion trajectories. Remove redundant trajectories and preserve one motion trajectory from all the similar motion trajectories. The representative object motion trajectory is approximated by Douglas-Pecker algorithm which is then stored in a database. There are many similarity measuring techniques exist i.e. distance-based similarity, feature-based similarity, probability-based similarity to retrieve a video based on their content. In this paper, we proposed motion trajectory similarity based video shot retrieval.

The rest of the paper is organized as follows: Section 2.1 describes System Representation. Section 2.2 describes Motion feature extraction and Motion trajectory. Section 2.3 proposes Trajectory comparison. Section 2.4 explains the Retrieval techniques. Section 2.5 discusses the delivery of video frames for small displays. Section 3 discusses the result of the experiment. Finally, Section 4 concludes the paper.

2. PROPOSED WORK FOR VIDEO CONTENT RETRIEVAL AND ADAPTIVE DISPLAY

2.1. System Representation

The block diagram of content-based video retrieval and display system is given in figure 1. The system consists of three modules: the input module, the query module, and display module.

In an Input Module, video shot is taken as input. The motion feature vector is computed for every frame of video shot by Exhaustive search algorithm.

Motion vectors only record the direction and magnitude of two consecutive frames so they are only comprised of local data that does not have much semantic meaning. Since motion vector is dispersed in the spatial domain, we construct motion trajectory from these vectors. To make use of the consistency of motion direction, the color distribution, and the overlapping area between macroblocks of two consecutive frames, all neighboring motion vectors are linked. These linked motion vectors form the so-called motion-flows or continual motion which contains more semantic meaning. Approximate these representative motion trajectories by generating control points and storing them in a database. **In a query module,** when a user wants to play any part of video shot, they must give a query video clip. Motion trajectory is identified for the query video clip. The query motion trajectory is compared with the database content by using similarity matching algorithm. If match occurs then relevant video shot is retrieved. **In a Display module,** in order to display in small devices, the retrieved relevant shot's foreground is extracted by *Region-growing* algorithm. In order to display in the small devices to feel the originality of video sequences, the foreground is rescaled and noisy irregular background is repaired by Temporal and Spatial filling-in method and finally, the reintegration of foreground and background of relevant video shot is done.

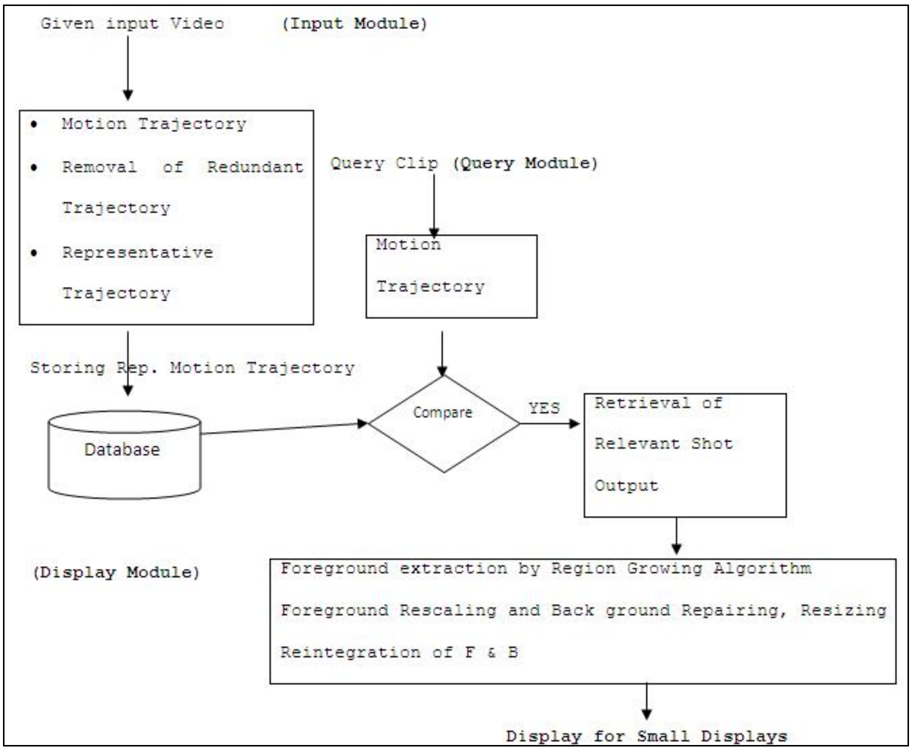


Fig. 1: Block-Diagram of Content-Based Video Retrieval and Delivery for Small Displays

2.2 Motion feature extraction and Motion trajectory:

Motion estimation is the process of determining motion vectors that describe the transformation from one frame to another; usually from adjacent frames in a video sequence. The idea behind block matching is to divide the current frame into a matrix of macro blocks. Each macro blocks are compared with corresponding block and its adjacent neighbours in the previous frame to create a vector that stipulates the movement of a macro block from one location to another in the previous frame. The search area for a good macro block match is constrained up to ‘p’ pixels on all fours sides of the corresponding macro block in previous frame. This ‘p’ is called as the search parameter. Larger motions require a larger search parameter. Usually the macro block is taken as a square of side 16 pixels, and the search parameter ‘p’ is 7 pixels. The matching of one macro block with another is based on the output of a cost function. The macro block that results in the least cost is the one that matches the closest to current block.

There are various cost functions, of which the most popular and less computationally expensive is Mean Absolute Difference (MAD) given in ‘eq1’.

$$MAD=1/N^2 \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} | C_{ij} - R_{ij} | \quad \text{------ ‘(1)’}$$

Where N is the side of the macro bock, C_{ij} and R_{ij} are the pixels being compared in current macro block and reference macro block, respectively. This algorithm calculates the cost function at each possible location in the search window. **Exhaustive Search** algorithm, also known as Full Search, is the most computationally expensive block matching algorithm. This algorithm calculates the cost function at each possible location in the search window. As a result of which it finds the best possible match.Several candidate local motions located in the previous frame could be connected to a local motion in the current frame. Thus, when we try to make the necessary connections between two local motion fields, each macroblock in the current frame may overlap with four macroblocks in the previous frame, as shown in Figure 2.

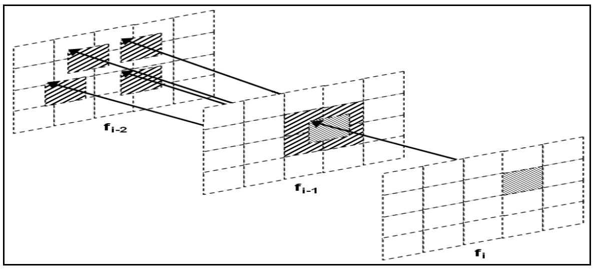


Fig. 2: A current motion vector and the four possible previous motion vectors that could be linked to it.

In other words, it is possible that several candidate local motions located in the previous frame could be connected with a local motion in the current frame. Therefore, two criteria were used in order to choose the most reliable candidate for the current local motion. The criteria are:

- Consistency of motion direction
- Color distribution

Suppose LMCur is a local motion and C denotes the set of candidates of LMCur in the previous frame. The most reliable candidate, LMancestor, is chosen based on the following rule given in ‘eq(2)’.

$$LM_{ancestor} = \underset{m€C}{\arg \mathrm{Min}} |(\theta_m, \alpha\phi_m)| \quad \text{--------------‘(2)’}$$

θ_m is the angle formed by LMCur and a possible candidate m and Φ_m is the color histogram difference between macroblocks. Furthermore, all the parameters are normalized so that all the values fall within the range [0,1], and α is weighting values that balance the influence of the two factors. Once the most reliable candidate of LMCur has been determined, *link the local motions* acquired at different time spots *to form a single motion trajectory* for each macro block. Approximate Representative Motion Trajectory is done by removing redundant motion trajectory. Since a large moving object may cover several similar motion trajectories, remove redundant trajectories and preserve one representative motion trajectory from all the similar motion trajectories.

Select a motion trajectory 'a' that has the longest duration in a shot. Suppose 'a' starts from time t_i and ends at time t_j, and let {B} be a set of motion trajectories whose start and end times are both within the duration of t_i and t_j. If the average of the relative spatial distance 'a' and 'b' is lower than a given threshold, remove a motion trajectory 'b', belonging to {B}. This is given in 'eq (3)'

$$\frac{\sum t \in [t_s, t_e] \, |(a_t - a_{ts}) - (b_{t} - b_{ts})|}{t_e - t_s} < \varepsilon \text{ -------'(3)'}$$

Where t_s and t_e are, respectively, the start and end times of motion trajectory 'a', a_t and b_t denote the spatial position of a and b at time't' respectively, a_{ts} and b_{ts} denote the spatial position of a and b at time t_s respectively. If the average of the relative spatial distance between 'a' and 'b' is lower than a given threshold ϵ, we consider b to be a sub segment of 'a'. As 'b' can be retrieved from 'a' by a partial matching process, it is redundant and can be removed. Consider 'a' as the representative of the motion trajectory removed from {B} and store it in the database.

To reduce time complexity and minimize the storage space required, remove redundant points from trajectory and select the necessary control points from the trajectory. In this paper, we used the Douglas - Peucker algorithm [15] recursively divides the line. Initially it is given all the points between the first and last point. It automatically marks the first and last point to be kept. It then finds the point that is furthest from the line segment with the first and last points as end points (this point is obviously furthest on the curve from the approximating line segment between the end points). If the point is closer than pre-set threshold to the line segment then any points not currently marked to keep can be discarded without the smoothed curve being worse than pre-set threshold.If the point furthest from the line segment is greater than 'ε' from the approximation then that point must be kept. This algorithm recursively calls itself with the first point and the worst point and then with the worst point and the last point (which includes marking the worst point being marked as kept).When the recursion is completed a new output curve can be generated consisting of all (and only) those points that have been marked as kept. Finally, the chosen intermediate points and the two end points are reserved as the control points of the trajectory.

$$EstDist_{ij}^{Q',D'} = |(Q'_j - Q'_i) - (D'_j - D'_i)|. \begin{pmatrix} 1\ 0 \\ 1\ 0 \\ 0\ 1 \\ 0\ 1 \end{pmatrix} \quad \text{---------- '(4)'}$$

Where i and j (i<j) denote the i^{th} and the j^{th} check points of two partial trajectories located on Q' and D', respectively. Q_j'- Q_i' and D_j'- D_i' represent the difference between the i^{th} and the j^{th} check points of Q' and D', respectively and $(Q'_j-Q'_i)-(D'_j-D'_i)$ is a 1x4 vector, and its subsequent part in (4) is a 4x2 matrix. Therefore the 1x2 vector is generated finally. The total distance metric between Q and D as given 'eq 5'.

$$Dist\ (Q,D) = \sum_{i=1}^{N-1} |EstDist_{i,i+1}^{Q',D'}| \quad \text{------------- '(5)'}$$

Cumulative length of the trajectory'd' value is used to find the similarity between Q and D in the spatial domain, as shown in Figure 3. $EstDist_{1,N}$ (Q',D') is greater than the predefined threshold value δ then the database trajectory D is not similar to Query trajectory Q. Once the value of $EstDist_{1,N}$ (Q',D') is less than the threshold value, Q' and D' can be divided into four sub trajectories by insert Q_2' and D_2'. Compute the sum of $EstDist_{1,2}$ (Q',D') and $EstDist_{2,N}$ (Q',D') as the distance between the sub trajectories. If the distance between two distinct sub trajectories is still larger than a predefined threshold δ, D will be filtered out. Otherwise, insert Q_3' and D_3' to further compute the distance between the sub trajectories. The process is executed repeatedly until the computed distance is larger than δ, or there are no more intermediate check points within each sub trajectory. All these comparison is made with the stored database. We quickly determine that trajectory D is similar to Q if the returned value of $|EstDist_{ij}^{Q',D'}|$ is less than the predefined threshold δ. If match occurs with database

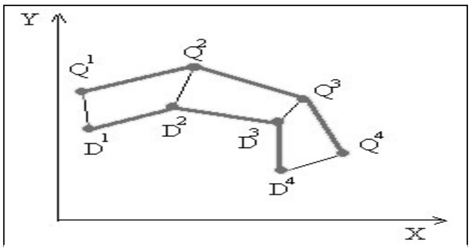


Fig. 3: Q and D in the spatial domain

information, the relevant shots are listed as output. As per user need, the required shot is selected from the relevant set and send that shot as input to the next section.

2.4 Displays for Small Displays:

2.4.1 Region-Growing Approaches:

The resulted retrieved shot from the previous step is assigned as input to the next stage. All the frames are converted into gray level first. The seed values and the threshold values are provided. Region-Growing approaches exploit in the foreground pixels, which are close together, have similar gray values otherwise it is a background pixels. This approach assumes an object-background image and picks a threshold that divides the image pixels into either object or background: i.e. X is a part of the object if $f(x) > T$, otherwise it is part of the background. This procedure group's pixels or sub-regions into larger regions based on a redefined criterion of growth. Start with a single pixel and add new pixels slowly. The algorithm is given below:

1. Choose the seed pixel.
2. Check the neighboring pixels and add them to the region if they are similar to the pixel.
3. Repeat step 2 for each of the newly added pixels; stop if no more pixels can be added.
4. More than one seeds grouped together in order to separate the background from the foreground of an image.

By selecting appropriate seed value and by setting a proper threshold, the region growing approach assumes an object-background image that divides the image pixels into either object or background. The background is segmented and the extracted foreground is shown in the figure 10. The frame is then smoothened using a Gaussian filter with a specified deviation σ to reduce noise. The local gradient $g(x,y) = [\ Gx^2 + Gy^2]^{1/2}$ and the edge direction $\alpha\ (x,y)=\tan^{-1}(Gx/Gy)$ are computed at each point. The ridge pixels are then threshold using two thresholds, T1, T2 with T1<T2. Ridge pixels with values greater than T2 are said to be stronger edge pixels and those values between T1 and T2 are said to be weak.

2.4.2 Foreground Rescaling and Background Repairing Algorithm:

To display the retrieved video sequences in the smaller devices such that cell phone, iPod etc., the foreground must be scaled down and the background should be repaired to feel the originality of video. Bilinear Interpolation determines the value of new pixels based on a weighted average of the pixels in the nearest 2*2 neighborhood in the original frame. This method is used to shrink the frame up to 50% of its originality.

In background repairing algorithm, the missing stationary background that is occlude by a stationary or moving object. For this, we first assign confidence values to each pixel in every frame. This value is deemed, to belong to the moving foreground or to the damaged area, is set to zero. The rest of the pixels are initialized to a confidence value of one. The process of background repairing is completed in two steps:

1. ***Temporal Filling-in:***

We search for the highest priority pixel location in the complete video sequence. Temporal information is copied from the temporally nearest undamaged location having the highest confidence.

2. ***Spatial Filling-in:***

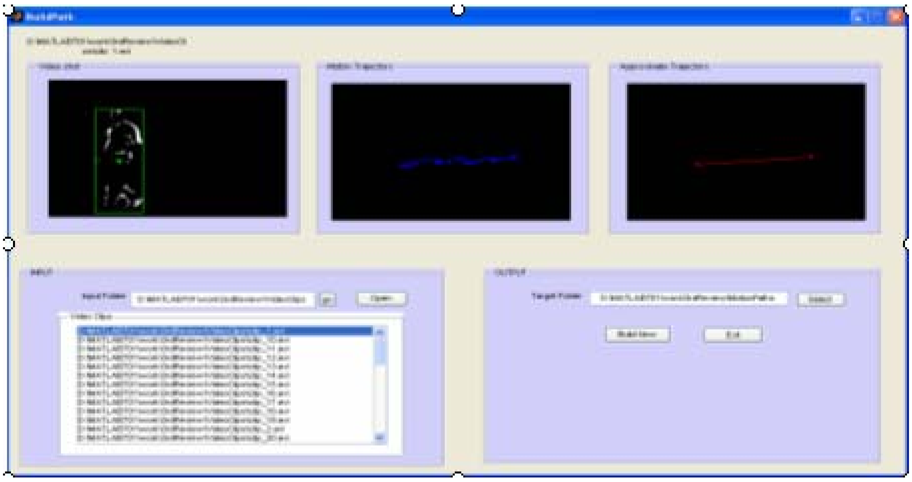
Once the temporal filling is over, we are left with a video sequence where all the frames have a hole at the same location. We again find the highest priority location to be filled-in and finding a best matching patch. This patch is copied to all the frames, so as to maintain consistent background throughout the sequence.

2.4.3 Reintegration of Foreground and Background:

The approach is to use a linear combination of the input images. The regions of high spatial variance across one image are computed by thresholding the intensity gradients, G=(GX, GY), for the horizontal and vertical directions using a simple forward difference. The regions of high temporal variance between two images are computed by comparing the intensity gradients of corresponding pixels from the two images. An importance image is computed a weighting function W by processing the gradient magnitude |G|. The weighted combination of input gradients gives us the gradient of the desired output. Reconstruct an image I from gradient field G and normalize pixel intensities in I is to closely match with the remaining intensities.

3. EXPERIMENTAL RESULTS:

This paper deals with how to display the retrieved relevant video shot on small devices such that cell phone, iPod etc. As discussed in section 2 all the steps are carried out.



Input Model: The motion trajectory is build for all video shots in the database. Motion trajectory axes which is constructed by the system and also shown the

Fig. 4: Building Motion Trajectory Framework

approximate trajectory for the simplified motion trajectory. Figure 4 shows the general block diagram of system framework. Figure 5 shows different Motion trajectory is constructed for different video shots.

Fig. 4(a): Construction of Motion Trajectory

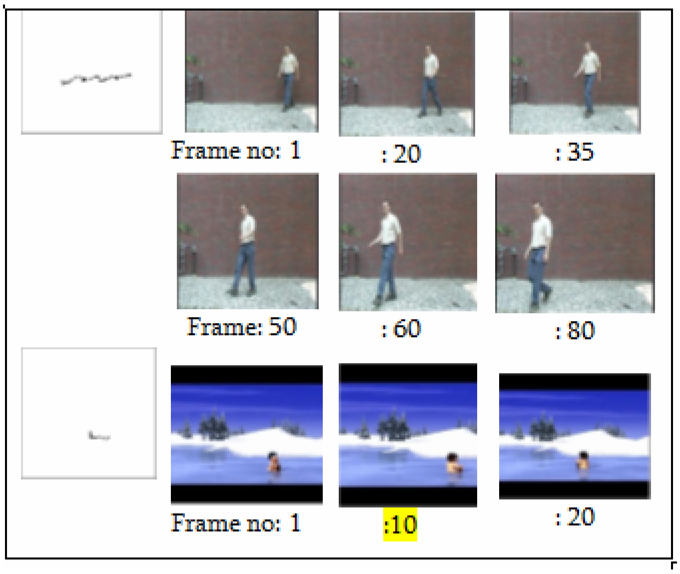


Fig. 5: Shows the approximate representative motion trajecto ry for different type of motion in order to store in the database.

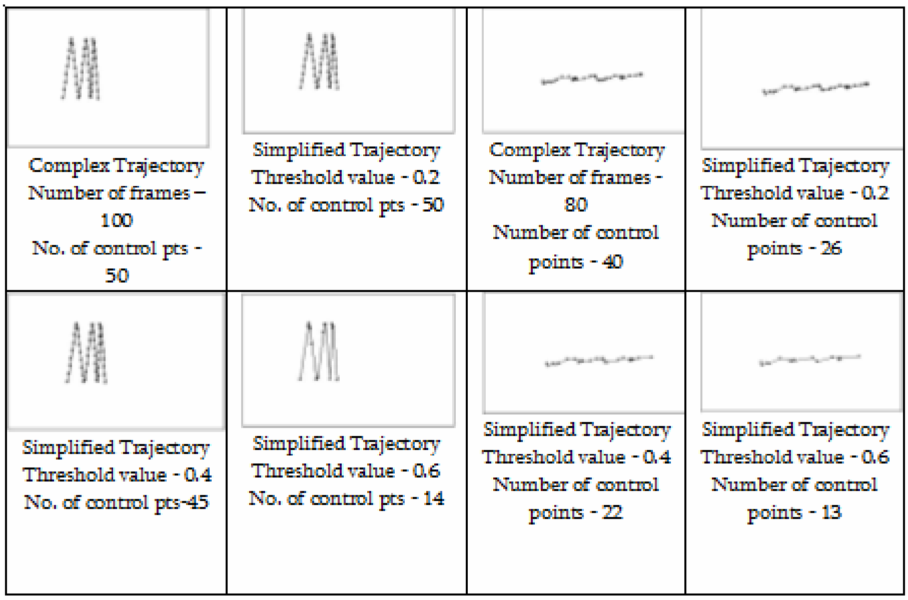
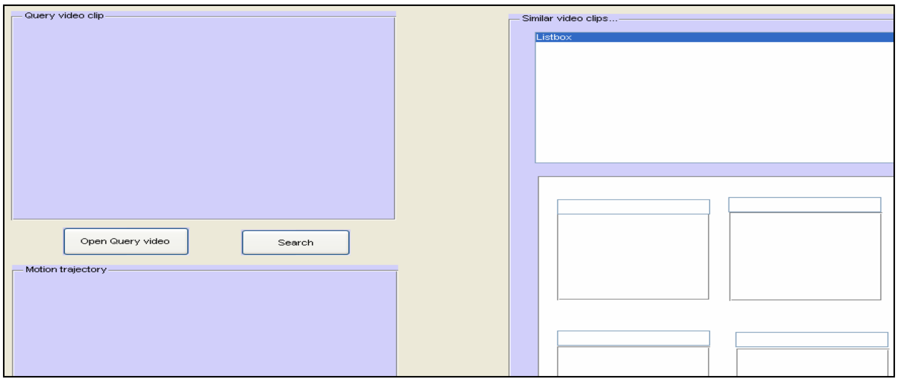


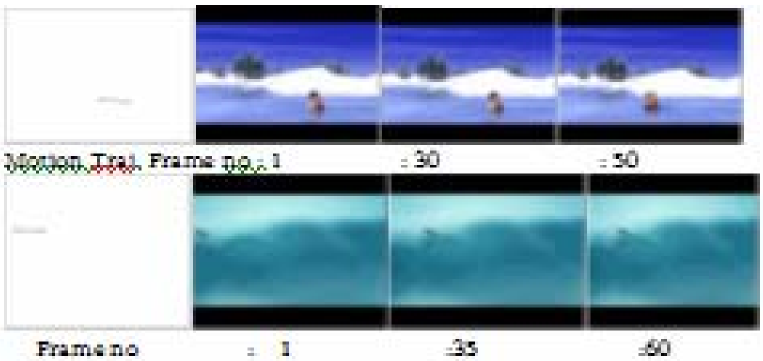
Fig. 5: Approximate Motion Trajectories for different Motion Type-
(a) Jumping (b) Horizontal

These approximated motion trajectories and its shot path are stored in the database for retrieval process.

The query video clip is given as input, system constructs the motion trajectory and Approximate it by using Douglas Pecker's algorithm. Fig. 6 shows the sample Output.



To evaluate the expressive power of the querying in the proposed system, measure the precision and recall of the system. Test cases are divided into four categories according to the motion within the shots. They are jumping, cyclic, horizontal and vertical motion.



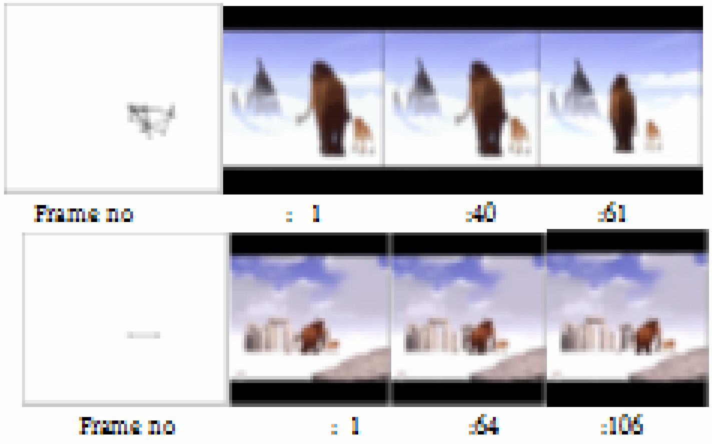


Fig. 6 (a) GuI design for (b) Query video clip (c) List of similar video shots (d) Possible Retrieval Video shot by comparing Motion trajectory.

Each category contained 10 to 20 shots. Construct a database of motion based video shots that contain 50 video shots. Each shot contains 5 to 200 frames. All the test cases had a frame sequence 320 * 240. Fig. 7 shows number of shots was considered for constructing the motion trajectory and its precision and recall evaluation.

Fig. 7(c) shows the average of recall and precision are changed due to the changes of threshold value in trajectory comparison. If the threshold value is 02 and 0.3 then the system will be able to retrieve the video shots which are much closer to the query shot. The system will be able to retrieve more similar trajectory video shots when the threshold value is 0.4 and 0.6.

Fig.8 (a) shows the number of shots missed, the shots which satisfy the user query are not returned, which satisfy the user query. Relevant video shots that have been retrieved by the system compared with the number of shots missed by the system which is shown in Fig. 8 (b). It has found that the system is efficient for retrieving similar video shots when the threshold value is 0.4 to 0.6.

Motion Category	No. Of Shots
Horizontal	20
Vertical	8
Jumping	10
Cyclic Movement	12

(a)

Threshold Value	Horizontal		Vertical	
	Recall	Precision	Recall	Precision
0.2	9/20	9/10	1/8	1/3
0.3	11/20	11/13	4/8	4/7
0.4	13/20	13/16	5/8	5/10
0.6	18/20	18/22	7/8	7/14
Threshold Value	**Jumping**		**Cyclic Movement**	
	Recall	**Precision**	**Recall**	**Precision**
0.2	3/10	3/5	2/12	2/3
0.3	5/10	5/7	3/12	3/6
0.4	6/10	6/9	6/12	6/9
0.6	7/10	7/13	9/12	9/13

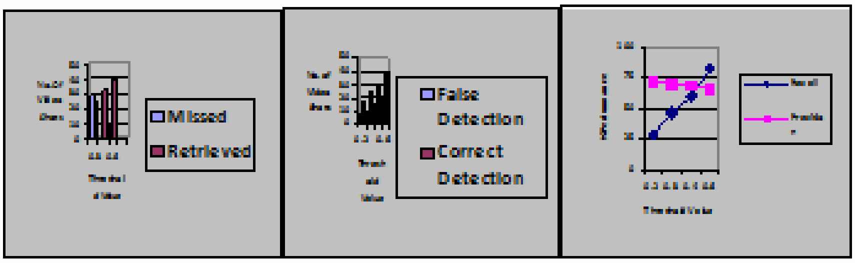
(b)

Threshold Value	Average			
	Recall		Precision	
	Decimal	%	Decimal	%
0.2	14/50= 0.2800	28	15/21=0.7142	71
0.3	23/50=0.4600	46	23/33=0.6971	70
0.4	30/50=0.6000	60	30/44=0.6818	68
0.6	41/50=0.8200	82	41/62=0.6612	66

(c)

Fig. 7: (a) Video Shots Under each Category (b) Recall & Precision Evaluation for each Category (c) Average precision and recall Evaluation.

Fig 8 (c) shows the relationship between threshold value and performance of the system. This shows that the average precision is low when the threshold value is 0.6 and average recall had a smooth increase of the recall measure as threshold value increases.



(a) (b) (c)

Figure 8(a) Missed Detection (b) False Detection (c) Performance compared with different Threshold value

Out of which these output relevant shots, the first few (one or two) shots are more precise shot which is send to very small device such as mobile phone, PDA etc. Fig. 9 shows the edge detection and the eliminated background is in different color in visible and the extracted foreground is displayed in Fig. 10. The resize of width and height are changed according to the user's preference and the reintegration of the foreground with the background is shown in Figure11. The final output of adaptive video shot is obtained which is now suitable to display in the small display and is shown in Figure12.

Fig. 9: Edge detection

Fig. 10: Foreground Extraction and Background Elimination

Figure11: Reintegration of Foreground & Background of Retrieved Shot

Figure 12: Resized Relevant Retrieved shots of video to display in small display

4. CONCLUSIONS AND FUTURE ENHANCEMENT

This paper addresses an effective retrieval and display of content based video for small displays. Adaptive Multimedia Content Retrieval and Delivery for small displays is one of the challenges faced by Multimedia Community. The proposed system is developed to retrieve a video shot by motion trajectories. This system needs video shots as the inputs whose motion vectors are extracted by using exhaustive search algorithm. This motion feature is linked across the consecutive frames of shot to form the motion trajectories. Remove redundant trajectories and preserve one motion trajectory from all the similar motion trajectories. The representative object motion trajectory is stored in a database. Query interface which allows users to search for similar video shots by giving query video clip.

Similarity matching algorithm is used to retrieve similar video shot from the database. In order to display those retrieved segmented shots in a small display, shape information of moving objects were extracted using Region- Growing algorithm. The segmented foreground was scaled down and re-integrated with the repaired and directly resized background to deliver effective adaptive video for small displays. Although it is already highly useful in most existing application scenarios, more flexible and economic methods are developed further. The challenging criteria is that synchronization original audio and video to display in the small displays.

In future, this could be enhanced by using multilevel and multimodal for various features of video. It would be allow searching a video shot for new query based on relevance feedback of user to get an efficient result.

REFERENCES:

1. I. Ahmed, X.Wei, Y.Sun and Y.Q.Zhang, (2005), Video Transcoding: an overview of various techniques and research issues, IEEE Trans. Multimedia, Vol.7(5), Oct.2005,pp. 793 -804.
2. J. Xin, C.W.Lin, and M.T. Sun, (2005), Digital Video Transcoding, Proc. IEEE vol 93(1), Jan. 2005,pp.84-97.
3. H.Knoche, J.D. MoCarthy and M.A.Sasse, (2005), Can small be beautiful? Assessing image resolution requirements for mobileTV, in Proc. 13th ACM Int. Conf. Multimedia (MM'05), 2005, pp. 829-838.
4. V.Setlur, S.Takagi, M. Gleicher, R. Ramesh and B. Gooch, (2004), Automatic Image Retargeting Comp. Sci. Dept, Northwestern Univ., Evanston, April 2004, Tech. Rep. NWU- CS-04-41.
5. F.Liu, and M. Gleicher, (2005), Automatic image retargeting with fisheye-view warping, in Proc. 18th ACM Symp. User Interface Technol.(UIST '05), 2005, pp. 153-162.
6. Yang Yongsheng and Lin Ming, (1999), A Survey on Content based video retrieval, Hong Kong University of Science & Technology.
7. C. V. Jawahar, BalaKrishna Chennupati, Balamanohar Paluri and Nataraj Jammalamadaka, "Video Retrieval Based on Textual Queries", in Proceedings of the Thirteenth International Conference on Advanced Computing and Communications, Coimbatore, December 2005.
8. Zuzana Cernekova and Ioannis Pitas, (2006), Information Theory-Based Shot Cut/Fade Detection and Video Summarization,IEEE Trans. Circuits Syst. Video Techn., Vol 16(1), Jan 2006,pp:82-91.
9. D.R. Chen, R.F. Chang,Y.L. Huang, (2000), Breast Cancer DiagnosisUsing Self-Organizing Map For Sonography, World Federation For Ultrasound in Med. & Biol., Vol. 26(3), 2000,.pp. 405–411.
10. C. Loizou, C. Christodoulou, C.S. Pattichis, R. Istepanian, M.Pantziaris, A. Nicolaides, (2002), Speckle Reduction in UltrasonicImages of Atherosclerotic Carotid

Plaque, 14th International IEEE Conference on Digital Signal Processing, 2002, pp.525-528,.

11. S. Pavlopoulos, E. Kyriacou, D. Koutsouris, K. Blekas, A. Stafylopatis, P. Zoumpoulis, (2000), Fuzzy Neural Network Computer Assisted Characterization of Diffused Liver Diseases Using Image Texture Techniques on Ultrasonic Images, IEEE Trans. Eng. in Medicine and Biology Magazine, Vol. 19(1), 2000, pp. 39-47.
12. C.C. Gotlieb and H.E. Kreyszig, (1990), Texture descriptorsbased on co-occurrence matrices, Comput. Vis. Graphics and Image Proc., Vol.51, 1990,pp 70-86.
13. (M. Rajpoot, (2002), Texture Classification Using Discriminant Wavelet Packet Subbands, 45th IEEE Midwest Symposium on Circuits and Systems, Tulsa-USA, 2002.
14. Y. Tao, V. Muthukkumarasamy, B. Verma, M. Blumenstein, (2003), A Texture Feature Extraction Technique Using 2D–DFT and Hamming Distance, 5th International Conference on Computational Intelligence and Multimedia Applications, Xi'an-China, 2003.
15. Chin-Wen Su, Hong-Yuan Mark Liao, Motion Flow-Based Video retrieval, IEEE Trans.Multimedia, vol.9,No.6, Oct. 2007,pp.1193-1201.
16. IoannisPatras,Emile Hendriks,ReginalL.Lagendijk, Probablistic Confidence Measures forBlock Matching Motion Estimation, IEEE Transaction on Circuits and Systems for Video Technology,vol.17(8), Aug. 2007, pp.989-995.