VGRAPH: An Effective Approach For Generating Static Video Summaries

Karim M. Mahmoud[1,2], Nagia M. Ghanem[1], and Mohamed A. Ismail[1]
[1] Computer and Systems Engineering Department
Faculty of Engineering, Alexandria University
Alexandria 21544, Egypt
[2] IBM, Egypt Branch
kmahmoud@eg.ibm.com

Abstract

A video summary is a sequence of still pictures that represent the content of a video in such a way that the respective target group is rapidly provided with concise information about the content, while the essential message of the original video is preserved. In this paper, we present VGRAPH, a simple yet effective video summarization approach that utilizes both color and texture features. This approach is based on partitioning the video into shots by utilizing the color features, and extracting video key frames using a nearest neighbor graph built from the texture features of the shots representative frames. Also, this paper introduces and illustrates an enhanced evaluation method based on color and texture matching. Video summaries generated by VGRAPH are compared with summaries generated by others found in the literature and the ground truth summaries. Experimental results indicate that the video summaries generated by VGRAPH have a higher quality than others.

1. Introduction

The digital revolution has brought many new applications and as a consequence research into new technologies that aim at improving the effectiveness and efficiency of video acquisition, archiving and indexing as well as increasing the usability of stored videos. This leads to the requirement of efficient management of video data such as video summarization.

A video summary is defined as a sequence of still pictures that represent the content of a video in such a way that the respective target group is rapidly provided with concise information about the content, while the essential message of the original video is preserved [11].

Over the past years, various video summarization approaches have been proposed. However one major drawback of these approaches is that they use a single visual descriptor such as the color features of the video frames; while other descriptors like texture is not considered. In this paper, we present VGRAPH, a simple yet effective approach for generating static video summaries that operates on video shots and utilizes both color and texture features. VGRAPH depends on partitioning the video sequence into shots based on the color features, extracting a representative frame from each shot, and then clustering the extracted shots representative frames using a nearest neighbor graph. This graph is built from the texture features of the shots representative frames. Also, we introduce an enhanced evaluation method that depends on color and texture features. VGRAPH approach is evaluated and compared with the state-of-the-art approaches for video summarization. Experimental results indicate that the video summaries generated by VGRAPH have a higher quality than others.

The rest of this paper is organized as follows. Section 2 introduces some related work. Section 3 presents VGRAPH approach and shows how to apply it to summarize a video sequence. Section 4 illustrates the evaluation method and reports the results of our experiments. Finally, we offer our conclusions and directions for future work in Section 5.

2. Related Work

A comprehensive review of video summarization approaches can be found in [14]. Some of the main approaches and techniques related to static video summarization which can be found in the literature are briefly discussed next.

In [5], a recursive multidimensional curve splitting algorithm is used to summarize of video sequences by small numbers of key frame. A video sequence is mapped to a curve trajectory in a high-dimensional space, using a feature vector carefully crafted so that the appearance of significant new information produces discontinuities or regions of high curvature in the trajectory. Then the algorithm is applied

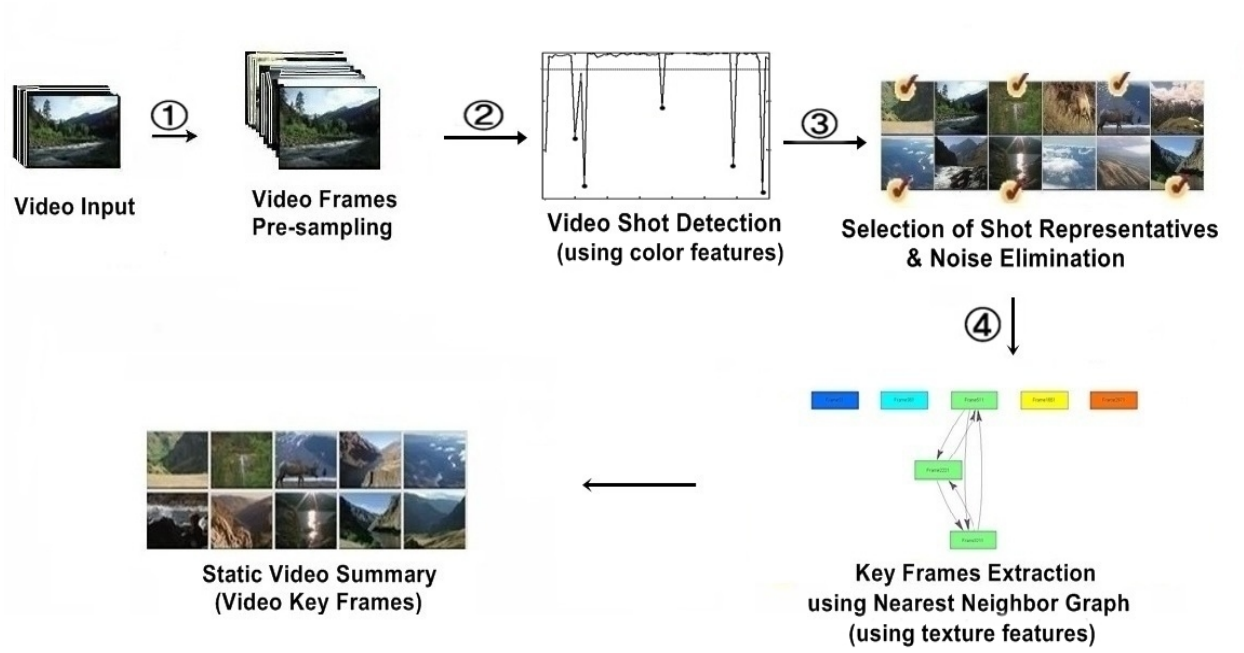

Figure 1. VGRAPH Video Summarization Approach

to recursively simplify the trajectory into constituent segments of low dimension. This method detects key frames that simplify the general trends of the video sequence and is less sensitive to local variations between consecutive frames than methods using frame differences or local filters. Finally, the set of the key frames are extracted from the video sequence which is concise but it contains sample frames from all the shots of the sequence.

In [10], an approach based on clustering the video frames using the Delaunay Triangulation (DT) is developed. The first step is pre-sampling the frames of the input video. Then, the video frames are represented by a color histogram in the HSV color space and the Principal Component Analysis (PCA) is applied on the color feature matrix to reduce its dimensionality. After that, the Delaunay diagram is built and clusters are formed by separating edges in the Delaunay diagram. Finally, for each cluster, the frame that is closest to its center is selected as the key frame. Finally, for each cluster, the frame that is closest to cluster's center is selected as the key frame. In this method, the quality of the video summaries by using three objective metrics: significance factor, overlap factor and compression factor. Although the proposed method has been designed to be fully automatic, it requires between 9 and 10 times the video length to produce the summary. Moreover, the method does not preserve the video temporal order.

In [6], an approach called STIMO (STIll and MOving Video Storyboard) is introduced. This approach is designed to produce on-the-fly video storyboards and it is composed of three phases. In the first phase, the feature vectors are extracted from the selected video frames by computing a color histogram in the HSV color space. In the second phase, a clustering method based on the Furthest-Point-First (FPF) algorithm is applied. To estimate the number of clusters, the pairwise distance of consecutive frames is computed using Generalized Jaccard Distance (GJD). If the distance is greater than a predefined threshold, the number of clusters is incremented. Finally, a post-processing step is performed for removing noise video frames. STIMO is evaluated using a comparison study with other approaches. In this study, a group of 20 people are asked to evaluate the produced summaries using the following procedure: the video is presented to the user, and then the corresponding summary is also shown. Then, the users are asked whether the summary is a good representation of the original video. The quality of the video summary is scored on a scale going from 1 (bad) to 5 (excellent), and the mean score is considered as an indication of the summary quality.

In [4], an approach called VSUMM (Video SUMMarization) is presented. In the first step, the video frames are pre-sampled by selecting one frame per second. In the second step, the color features of video frames are extracted from Hue component only in the HSV color space. In the third step, the meaningless frames are eliminated. In the fourth step, the frames are clustered using k-means algorithm where the number of clusters is estimated by computing the pairwise Euclidean distances between video frames and a key frame is extracted from each cluster. Finally, an-

other extra step occurs in which the key frames are compared among themselves through color histogram to eliminate that similar key frames in the produced summaries.

3. VGRAPH Approach

Figure 1 shows the steps of VGRAPH approach to generate static video summaries. First, the original video is pre-sampled (step 1). Second, the pre-sampled video is segmented into shots using the color features (Step2). Third, noise frames are eliminated and a representative frame is selected from each shot (Step 3). In step 4, the key frames are extracted using nearest neighbor graph which is built from the texture features extracted from shots representative frames. These steps are explained in details in the following subsections.

3.1. Video Frames Pre-sampling

The target of the pre-sampling step is to reduce the number of frames to be processed. Selecting a suitable sampling rate is very important; as a low sampling rate leads to poor video summaries; while a large sampling rate shortens the video summary. According to our experimental tests, the sampling rate used in VGRAPH approach is selected to be one frame per second. So, for a video sample of duration one minute, and a frame rate of 30 fps (i.e., 1800 frames); the number of extracted frames is 60 frames.

3.2. Temporal Video Segmentation

An important issue is to determine what base temporal unit the extracted key frame represents. In video summarization approaches, key frames can be extracted to represent individual shots as an instant step or directly represent the video clip. The first method is often classified as shot-based and requires the detection of the shot boundaries. The second approach, on the other hand, may proceed without the knowing the shot boundaries and is called clip-based [14]. In VGRAPH approach, the key frames extraction process is a shot-based method, which requires video segmentation by detecting the shot boundaries.

For video temporal segmentation step, a simple shot boundary detection method [7] is used. In this method the pairwise distances of consecutive frames are computed from color features in the extracted sample and every time the distance between two consecutive frames exceeds threshold T, a shot is created.

In VGRAPH, the color features are extracted using the color histogram computed from the HSV (Hue-Saturation-Value) color space using 32 bins of H, 4 bins of S, and 2 bins of V. The HSV color space is used because it is developed to provide an intuitive representation of color and to be near to the way in which humans perceive color. The quantization of the color histogram is established through experimental tests [9] and aims at reducing the amount of data without losing important information.

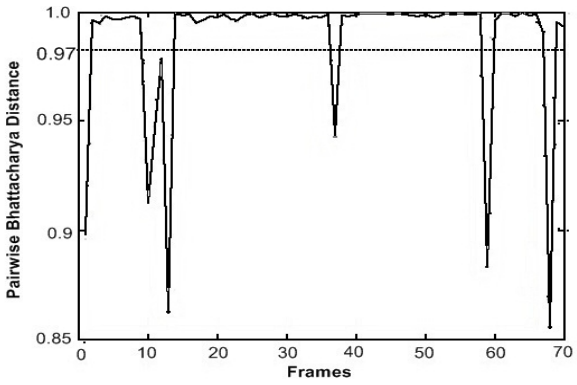

Figure 2. Pairwise Bhattacharyya distances of consecutive frames of video "The Great Web of Water,segment 02", available at Open Video Project [1]

In VGRAPH, the Bhattacharyya distance [8] is used to compute the pairwise distances between the consecutive frames. The Bhattacharyya distance between two histograms P and Q of size n; is defined as:

$$BhattacharyyaDistance = \sum_{i=0}^{n} \sqrt{\sum Pi \bullet \sum Qi} \quad (1)$$

To detect video shots, the pairwise Bhattacharyya distances between consecutive frames are computed, if the Bhattacharyya distance between two consecutive frames is less than threshold T, a shot is created. The threshold value applied in this work is equal to 0.97 (established through experimental tests). Figure 2 shows the pairwise Bhattacharyya distances of sampled frames of the video "The Great Web of Water, segment 02" (video is available at Open Video Project [1]).

Using the Bhattacharyya distance as dissimilarity measure has many advantages [2]. First, the Bhattacharyya measure has self-consistency Property as all poisson errors are forced to be constant therefore ensuring the minimum distance between two observations points is indeed a straight line. Second advantage is the independency between Bhattacharyya measure and bin widths, as the Bhattacharyya metric the contribution to the measure is the same irrespective of how the quantities are divided between bins. Therefore the Bhattacharyya statistic is unaffected by the distribution of data across the histogram and is the only form of sum-of-product functions with this property. Finally, the Bhattacharyya measure is dimensionless, as it is not affected by the measurement scale used, when Bhattacharyya measure is used to compare two identical distributions, it has been proven that the term is maximized to a value of one [2].

3.3. Noise Elimination and Selection of Shots Representative Frames

After video segmentation step, those shots of size 1 frame only are neglected as they are considered noise frames according to experimental tests, then for each shot we select a a representative. In this approach, we select the second frame as a shot representative. Although the frames grouped in the same shot are similar; the first and the last frames contain a fade effect due to transitions between different shots.

3.4. Key Frames Extraction using Nearest Neighbor Graph

The k-nearest neighbor graph (k-NNG) is a graph in which two vertices p and q are connected by an edge, if the distance between p and q is among the k-th smallest distances from p to other objects from p. A Nearest neighbor graph (NNG) is k-nearest neighbor graph (k-NNG) with k value equals to 1. In VGRAPH approach, we build a nearest neighbor graph for the shots representative frames extracted from the previous step using texture features and then the key frames are extracted from the graph as it will be illustrated later. According to our experimental tests, we found that using texture features is more effective than using color features in order to cluster the shot shots representative frames in this step; as using the color histogram in this step sometimes gives false similarity detection for completely different frames. This is different than using the color features to detect video shots as done in the previous step, where the consecutive frames are temporally correlated and similar to each other within the same video shot.

In VGRAPH, the texture features of the shots representative frames are extracted using Discrete Haar Wavelet Transforms [12], because it is fast to compute and also have been found to perform well in practice. Each frame is converted into HSV color space and its size is reduced into 64X64 pixels in order to reduce computation without losing significant image information. Next step, is applying a two-dimensional Haar Wavelet transform on the reduced HSV image data with decomposition level 3. Finally, the texture features of the video frames are extracted from the approximation coefficients of the Haar Wavelet Transforms.

The Haar wavelet's mother wavelet function $\psi(t)$ can be described as:

$$\psi(t) = \begin{cases} 1, & 0 \leq t \leq 0.5 \\ -1, & 0.5 \leq t \leq 1 \\ 0, & \text{otherwise} \end{cases}$$

and its scaling function $\phi(t)$ can be described as:

$$\phi(t) = \begin{cases} 1, & 0 \leq t \leq 1 \\ 0, & otherwise \end{cases}$$

After extracting the texture features of the shots representative frames, a nearest neighbor graph is built using the Bhattacharya distance as a dissimilarity measure. In this graph, two frames are connected only if they are texture-based similar, in this case the Bhattacharya distance between the texture features is equal or greater than threshold *EpsTexture* = 0.97 (threshold value established through experimental tests). Then, a reverse nearest neighbor graph (RNNG) is constructed from the previously built NNG graph. To extract the key frames, each strongly connected component is identified as a cluster in the RNNG graph and a frame is selected from each cluster.

Following are some notations and definitions to illustrate the clustering of frames using nearest neighbor graph:
X: video frames dataset
$\mathbf{x}_i$, $\mathbf{x}_j$: ith, jth frames in X
p, q: any two frames in X
$\mathbf{d}_{ij}$: Bhattacharyya distance between two frames (x_i, x_j)
***EpsTexture*-NN(p):** *EpsTexture*-Nearest neighbor frame of frame p
***EpsTexture*-RNN(p):** *EpsTexture*-Reverse nearest neighbor frame set of frame p
RNNG: Reverse nearest neighbor graph
SCC: Strongly connected component

Definition 1. ***(EpsTexture-NN) EpsTexture-Nearest Neighbor Frame:*** *EpsTexture-NN(x_j) is defined as $x_i|d_{ij}$ = largest Bhattacharya distance from x_j, where $d_{ij} >=$ EpsTexture threshold value. For a given frame x_j, after sorting all the Bhattacharya distances from x_j to the remaining frames in X, the frame with largest Bhattacharya distance in X is the nearest neighbor Frame of x_j, on condition that the distance is greater than or equal to EpsTexture threshold value.*

Definition 2. ***(EpsTexture-RNN) EpsTexture-Reverse Nearest Neighbor Frames Set:*** *EpsTexture-RNN(x_i) is defined as $\{x_j | x_j \in X,\ x_i = EpsTexture\text{-}NN(x_j)\}$, the set of all frames x_j that consider x_i as their EpsTexture-nearest neighbor frame.*

Definition 3. ***(SCC) Strongly connected component*** *A Strongly Connected Component (SCC) of a graph partitions the frames vertices into subsets wherein all frames in a subset are mutually reachable.*

The steps involved in VGRAPH clustering algorithm are as follows:

1. For each detected shot representative frame p, calculate the EpsTexture-NN(p).
2. Calculate the EpsTexture-Reverse nearest neighbor frames set (EpsTexture-RNN) and build corresponding graph.

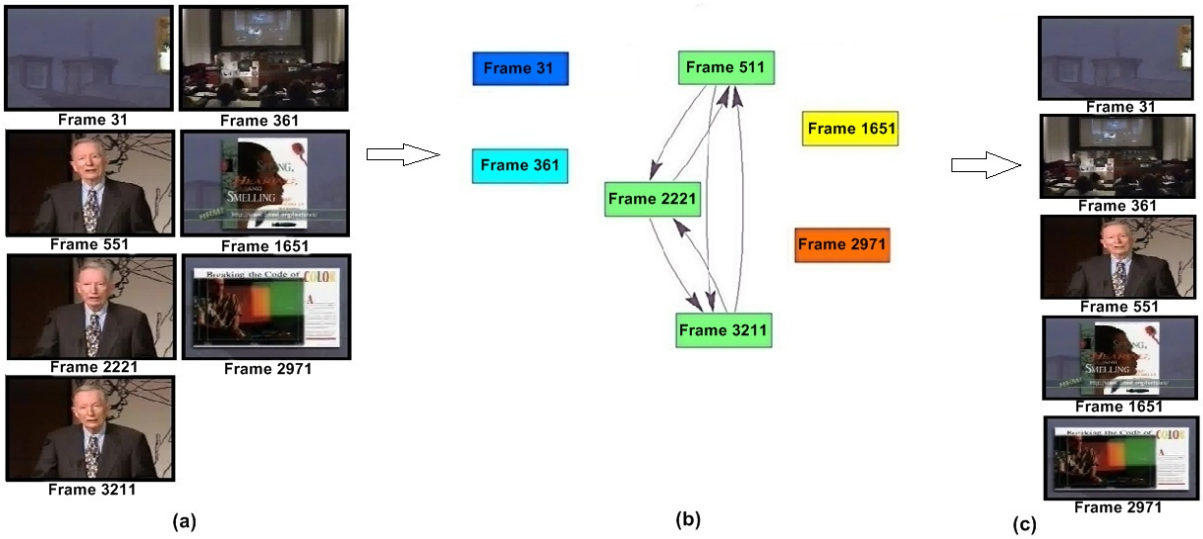

Figure 3. Steps of VGRAPH applied on video "Senses and Sensitivity, Introduction to Lecture 2 (video is available at Open Video Project [1]) (a) Shots Representative Frames (b) Eps-Nearest Neighbor Graph (c) Extracted Key Frames

3. Find all Strongly connected components (SCC) in EpsTexture-RNN graph

To find the strongly connected components in the graph, we use Tarjans Algorithm [13] where each strongly connected component is considered a video cluster. After clustering the video frames, the final step is selecting the key frames from the video clusters. For each cluster the first frame in the ordered frames sequence is selected as a key frame to construct the video summary. According to our experiments, we found that this first frame usually is the best representative of the video cluster to which it belongs.

Figure 3 shows the steps of key frames extraction of the video "Senses and Sensitivity, Introduction to Lecture 2" (video is available at Open Video Project [1]), where every strongly connected component is highlighted with different color, the output clusters of this example are as follows:
Cluster 1: {Frame31}
Cluster 2: {Frame361}
Cluster 3: {Frame511, Frame2221, Frame3211}
Cluster 4: {Frame1651}
Cluster 5: {Frame2971}

The final step is selecting the first frame from each cluster as a key frame, the final extracted key frames set is: {Frame31, Frame361, Frame511, Frame1651, Frame2971}.

4. Experimental Evaluation

In this paper, an enhanced evaluation method described in [9] is used to evaluate the quality of video summaries. In this method, the video summary is built manually by a number of users from the sampled frames and the user summaries are taken as reference (i.e. ground truth) to be compared with the automatic summaries obtained by different methods [4].

The enhancements proposed to the evaluation method aims at providing a more perceptual assessment of the quality of the automatic video summaries where different summaries are compared using both color and texture features. The color and texture features are extracted as illustrated in previous sections. In this method, every two frames in the automatic summary and the user summary are compared to each other, once two frames are color-based similar and texture-based similar, they are excluded from the next iteration of the comparison process. Also, the Bhattacharya distance is used to detect both color and texture similarity; in this case the distance threshold value for color and texture similarity is set to 0.97 for each. Figure 4 shows the flowchart of the proposed enhanced evaluation method. Following is the pseudo code for determining the color-based matched frames:

Algorithm 4.1: ISCOLORMATCHED(I, J)

comment: I & J are two input frames

$iRGB \leftarrow ReadRGB(I)$
$jRGB \leftarrow ReadRGB(J)$
$iHSV \leftarrow ConvertRGBToHSV(iRGB)$
$jHSV \leftarrow ConvertRGBToHSV(jRGB)$
$iHist \leftarrow ComputeHistogram(iHSV, H : 32, S : 4, V : 2)$
$jHist \leftarrow ComputeHistogram(jHSV, H : 32, S : 4, V : 2)$
$distance \leftarrow ComputeBhattacharyya(iHist, jHist)$
if $distance \geq 0.97$
  then return (**true**)
  else return (**false**)

Following is the pseudo code for determining the texture-based matched frames:

Algorithm 4.2: ISTEXTUREMATCHED(I, J)

comment: I & J are two input frames

$iRGB \leftarrow ReadRGB(I)$
$jRGB \leftarrow ReadRGB(J)$
$iRdRGB \leftarrow ReduceSize(iRGB)$
$jRdRGB \leftarrow ReduceSize(jRGB)$
$iHSV \leftarrow ConvertRGBToHSV(iRdRGB)$
$jHSV \leftarrow ConvertRGBToHSV(jRdRGB)$
$iCoef \leftarrow GetCoefficientsHaarDWT(iHSV, level : 3)$
$jCoef \leftarrow GetCoefficientsHaarDWT(jHSV, level : 3)$
$distance \leftarrow ComputeBhattacharyya(iCoef, jCoef)$
if $distance \geq 0.97$
  then return (**true**)
  else return (**false**)

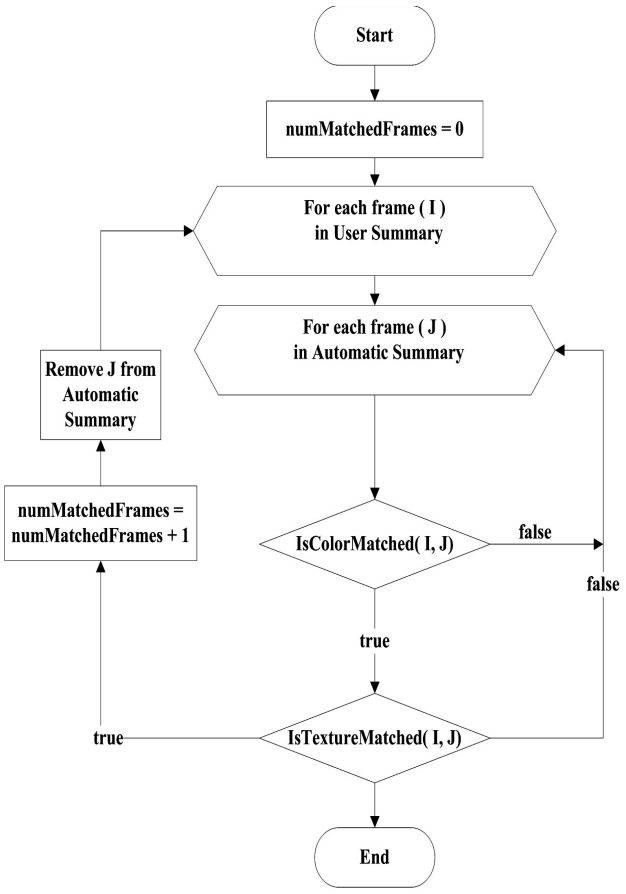

Figure 4. Flowchart of the enhanced evaluation method.

Approach	F-Measure
OV [5]	0.67
DT [10]	0.61
STIMO [6]	0.65
VSUMM [4]	0.72
VGColor	0.72
VGRAPH	**0.75**

Table 1. Mean F-measure achieved by different video summarization approaches.

In order to evaluate the automatic video summary, the F-measure is used as a metric. The F-measure consolidates both Precision and Recall values into one value using the harmonic mean [3], and it is defined as:

$$F\text{-}measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (2)$$

The Precision measure of video summary is defined as the ratio of the total number of color-based matched frames and texture-based matched frames to the total number of frames in the automatic summary; and the Recall measure is defined as the ratio of the total number of color-based matched frames and texture-based matched frames to the total number of frames in the user summary

Figure 5 shows the details of applying our proposed evaluation method on video 'America's New Frontier - segment 04', the number of frames in the automatic summary is 8 frames, the number of frames in user summary is 7 frames, and the number of matched frames are 6 frames. The F-measure for this case is calculated as follows: Presicion = $\frac{6}{8}$, Recall = $\frac{6}{7}$, so the F-measure = 0.79.

VGRAPH approach is evaluated on a set of 50 videos selected from the Open Video Project [1] . All videos are in MPEG-1 format (30 fps, 352 240 pixels). These videos are distributed among several genres (documentary, historical, lecture, educational) and their duration varies from 1 to 4 min. Also, we used the same user summaries used in [4, 9] as a ground-truth data. The user summaries were created by 50 users, each one dealing with 5 videos, meaning that each video has 5 summaries created by five different users. So, the total number of video summaries created by the users is 250 summaries and each user may create different summary.

For comparing VGRAPH approach with other approaches, we used the results reported by three approaches: VSUMM [4], STIMO [6], and DT [10]. In addition to that, the automatic video summaries generated by our approach were compared with the OV summaries generated by the algorithm in [5]. All the videos, user summaries, and automatic summaries are available publicly [1].

[1] http://sites.google.com/site/vgraphsite/

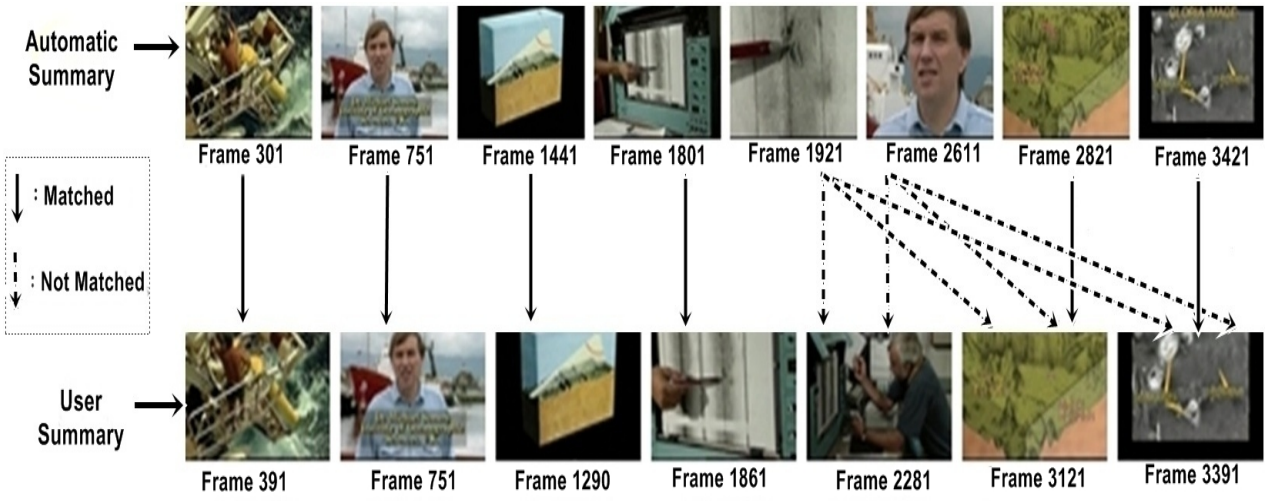

Figure 5. Evaluation of automatic summary of video "America's New Frontier - segment 04", available at Open Video Project [1]

Table 1 shows the mean F-measure achieved by the different video summarization approaches. The results indicate that VGRAPH performs better than all other approaches.

In addition to previous approaches, we implemented a video summarization approach called VGColor using the same steps as VGRAPH, but we used the color features to build nearest neighbor graph instead of using texture features. The color features extraction method used in VGColor is the same used in VGRAPH's temporal video segmentation step. The reason for implementing VGColor is to test the effect of using color instead of texture to build nearest neighbor graph. As it is shown in Table 1, VGRAPH gives better results than VGColor. According to our experimental tests, using texture features is more effective than using color features in order to cluster the shot shots representative frames in this step; as using the color histogram in clustering step sometimes gives false similarity detection for completely different frames. Unlike using the color features in the temporal video segmentation step to detect video shots, where the consecutive frames are temporally correlated and similar to each other within the same video shot.

5. Conclusion

In this paper, we presented VGRAPH, a simple yet effective approach for generating static video summaries. This approach is based on partitioning the video into shots utilizing the color features, clustering the extracted shot representatives, and then selecting the key frames. In order to cluster these shots representatives, a nearest neighbor graph is built from the texture features of each shot representative frame.

As an additional contribution, we introduced an enhanced evaluation method based on color and texture matching. The main advantage of this evaluation method is to provide a more perceptual assessment of the quality of automatic video summaries.

Future work includes conducting experiments on specific types of videos like: sports videos and videos recorded from surveillance systems. Also, another interesting future work could be generating video skims (dynamic key frames, e.g. movie trailers) from the extracted key frames. Since the video summarization step is considered as a prerequisite for video skimming [14], the extracted key frames from VGRAPH can be used to develop an effective video skimming approach.

References

[1] Open video project. [online]. available: http://www.open-video.org.

[2] F. J. Aherne, N. A. Thacker, and P. I. Rockett. The Bhattacharyya metric as an absolute similarity measure for frequency coded data. *Kybernetika*, 34(4):363–368, 1998.

[3] H. M. Blanken, A. De Vries, H. E. Blok, and L. Feng. *Multimedia retrieval*. Springer Verlag, 2007.

[4] S. E. F. de Avila, A. P. B. Lopes, et al. VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern Recognition Letters*, 32(1):56–68, 2011.

[5] D. DeMenthon, V. Kobla, and D. Doermann. Video summarization by curve simplification. In *Proceedings of the sixth ACM international conference on Multimedia*, pages 211–218. ACM, 1998.

[6] M. Furini, F. Geraci, M. Montangero, and M. Pellegrini. STIMO: STIll and MOving video storyboard for the web scenario. *Multimedia Tools and Applications*, 46(1):47–69, 2010.

[7] Guimaraes, S. J. F., M. Couprie, A. de Albuquerque Araujo, and N. Jeronimo Leite. Video segmentation based on 2D image analysis. *Pattern Recognition Letters*, 24(7):947–957, 2003.

[8] T. Kailath. The divergence and Bhattacharyya distance measures in signal selection. *Communication Technology, IEEE Transactions on*, 15(1):52–60, 1967.

[9] K. M. Mahmoud, M. A. Ismail, and N. M. Ghanem. VSCAN: An Enhanced Video Summarization Using Density-Based Spatial Clustering. In *Image Analysis and Processing Conference-ICIAP 2013*, volume 1, pages 733–742. Springer Berlin Heidelberg, 2013.

[10] P. Mundur, Y. Rao, and Y. Yesha. Keyframe-based video summarization using Delaunay clustering. *International Journal on Digital Libraries*, 6(2):219–232, 2006.

[11] S. Pfeiffer, R. Lienhart, S. Fischer, and W. Effelsberg. Abstracting digital movies automatically. *Journal of Visual Communication and Image Representation*, 7(4):345–353, 1996.

[12] R. S. Stanković and B. J. Falkowski. The Haar wavelet transform: its status and achievements. *Computers & Electrical Engineering*, 29(1):25–44, 2003.

[13] R. Tarjan. Depth-first search and linear graph algorithms. *Journal of Visual Communication and Image Representation*, 1(2):146–160, 1972.

[14] B. T. Truong and S. Venkatesh. Video abstraction: A systematic review and classification. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 3(1):3, 2007.