A Static Video Summarization Method Based on Hierarchical Clustering

Silvio Jamil F. Guimarães and Willer Gomes

Audio-Visual Information Processing Laboratory (VIPLAB)
Institute of Informatics - Pontifícia Universidade Católica de Minas Gerais
Belo Horizonte, MG, Brazil
{sjamil, willer}@pucminas.br
<http://www.viplab.inf.pucminas.br>

Abstract. Video summarization is a simplification of video content for compacting the video information. The video summarization problem can be transformed to a clustering problem, in which some frames are selected to saliently represent the video content. In this work, we use a graph-theoretic divisive clustering algorithm based on construction of a minimum spanning tree to select video frames without segmenting the video into shots or scenes. Experimental results provides a visually comparison between the new approach and other popular algorithms from the literature, showing that the new algorithm is robust and efficient.

Keywords: Video summarization, Minimum spanning tree, Video analysis.

1 Introduction

The increasing number of video files has become the task of searching a specific content very expensive, because it is necessary to index the video information. Usually, there are two approaches to cope with the index problem: (i) manual notation; and (ii) automatic notation. The former is expensive and subjective, since it depends on the experts to perform this notation. The second one is objective and is directly related to the visual contents, however it depends on the features which are used to index. The cost to find a specific content related to a video depends on the size of the index, thus instead of considering all video content, we summarize it in order to reduce the search space. In literature, there are many approaches to simplify the video content [8,9,11,7,1,6,3,2]. Thus, video summarization is a simplification of video content for compacting the video information, also the feature or similarity measure used to this simplification depends on the application. The video summarization problem can be transformed to a clustering problem, in which some frames are selected to saliently represent the video content, as illustrated in Fig. 1. In [1] was proposed an approach to cope with the video summarization problem in which the clustering is achieved by a k-means algorithm, but it is necessary to know a priori the number of clusters. In [2] was proposed the use of graph-theoretic FCM algorithm for video summarization, however the graph creation is directly related to number of centers. In [6] it was used a Delaunay triangulation to automatically identify the frame clusters, however this approach is expensive and produces very compressed summaries. VISTO [3] is based on low-level video frames color

Fig. 1. Steps for video summarization (adapted from [6])

feature extraction and on a modification of furthest point-first algorithm to cluster the frames. This approach is fast but the summaries is big.

In this work, we use a graph-theoretic divisive clustering algorithm based on construction of a Minimum Spanning Tree (MST) to select video frames without segmenting the video into shots or scenes, i.e., our approach eliminates pre-processing steps. It is important to note that according to [4] the MST approach for clustering is hierarchical, and thanks to this property, it is easy to compute a video summary regarding the specified number of keyframes.

The paper is organized as follows. Section 2 describes the clustering problem using minimum spanning tree and, also, it is defined many concepts used in our work. Section 3 describes our methodology to solve the video summarization problem. Section 4 describes the performed experiments together to a comparative analysis between our approach and the others methods. Finally, we give some conclusions in Section 5.

2 Clustering by Minimum Spanning Tree Approach

Let $\mathbb{A} \subset \mathbb{N}^2$, $\mathbb{A} = \{0, \dots, H-1\} \times \{0, \dots, W-1\}$, where H and W are the width and height of each frame, respectively, and, $\mathbb{T} \subset \mathbb{N}$, $\mathbb{T} = \{0, \dots, N-1\}$, in which N is number of frames of a video.

Definition 1 (Frame). A frame f is a function from $\mathbb{A}$ to $\mathbb{Z}$, where for each spatial position (x, y) in $\mathbb{A}$, $f(x, y)$ represents the color value at pixel location (x, y) .

Definition 2 (Video). A video V_N , in domain $\mathbb{A} \times \mathbb{T}$, can be seen as a sequence of frames f . It can be described by

$$V_N = (f)_{t \in \mathbb{T}} \quad (1)$$

where N is the number of frames contained in the video.

Definition 3 (Frame similarity). Let f_{t_1} and f_{t_2} be two video frames at locations t_1 and t_2 , respectively. Two frames are similar if a distance measure $\mathcal{D}(f_{t_1}, f_{t_2})$ between them is smaller than a specified threshold (δ) . The frame similarity is defined as

$$FS(f_{t_1}, f_{t_2}, \delta) = \begin{cases} 1, & \text{if } \mathcal{D}(f_{t_1}, f_{t_2}) \leq \delta \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

There are several choices for global measures $\mathcal{D}(f_{t_1}, f_{t_2})$, i.e., the distance measure between two frames, e.g. histogram/frame difference, histogram intersection, difference of histograms means, and others. After selecting one, it is possible to construct a frame similarity graph based on a video V_N and a distance measure as follows.

Definition 4 (Frame similarity graph – G^δ). Let V_N be a video with N frames. A frame similarity graph $G^\delta = (N, E^\delta)$ is a weighted undirected graph. Each node $v_{t_1} \in N$ represents a frame $f_{t_1} \in V_N$. There is an edge $e \in E^\delta$ with weight $\mathcal{D}(f_{t_1}, f_{t_2})$ between two nodes v_{t_1} and v_{t_2} if frame similarity of associated frames is equal to 1:

$$E^\delta = \{ (v_{t_1}, v_{t_2}, \mathcal{D}(f_{t_1}, f_{t_2})) \mid v_{t_1} \in N, v_{t_2} \in N, FS(f_{t_1}, f_{t_2}, \delta) = 1 \} \quad (3)$$

Fig. 2(a) illustrates a frame similarity graph of a real video in which only the frames 1, 501, 1001, 1501, 2001, 2501 and 3001 are sampled. The similarity measure used is the histogram intersection of HSV color space and is in the range $[0, 100]$.

Fig. 2. Examples of video summarization. (a) frame similarity graph; (b) minimum spanning tree of frames; (c) deleting edge with weight equals to 72; (d) deleting edge with weight equals to 49.

In order to perform the video summarization, without considering a video partitioning (or segmentation) step, we propose the use of a divisive clustering algorithm. According to [4], the best-known graph-theoretic divisive clustering algorithm is based on construction of the minimum spanning tree of the data [10]. In this work, we define the minimum spanning tree of frames as a graph structure that preserves the video content and the relationship between all video scenes.

Definition 5 (Minimum spanning tree of frames - $fMST_{G^\delta}$). Let $G^\delta = (N, E^\delta)$ be a frame similarity graph. The minimum spanning tree of frames $fMST_{G^\delta} = (N, E_1^\delta)$ is a subgraph of G^δ that minimizes the sum of weights of the edges E^δ .

According to [5], a k -clustering divides the elements into k non-empty groups, in which the insertion of an element into a group depends on distance measure between this element and the elements already in the group. In order to compute the video summarization a k -clustering divides the video sequence into k video scenes, and consequently, it is necessary to eliminate $k - 1$ edges from the MST. To follow, the ordered edge sequence and the value transform are defined in order to simplify the frame clustering algorithm.

Definition 6 (Ordered edge sequence - $S^{E_1^\delta}$). Let $fMST_{G^\delta}$ be the minimum spanning tree of frames. Let $E_1^\delta = \{e = (a, b, c)\}$ be the collection of weight edges of $fMST_{G^\delta}$. Let $S_i^{E_1^\delta}$ be the i -th edge $e_i = (a_i, b_i, c_i) \in E_1^\delta$ between two frames a_i and b_i with weight c_i . The ordered edge sequence ($S^{E_1^\delta}$) with respect to the weight is define by $S^{E_1^\delta} = (S_i^{E_1^\delta})_{i \in [1, N]}$ in which $S_i^{E_1^\delta} \leq S_{i+1}^{E_1^\delta}$ if $c_i \leq c_{i+1}$.

Definition 7 (Value transform - $T(\Delta)$). Let $fMST_{G^\delta}$ be the minimum spanning tree of frames. Let $S^{E_1^\delta}$ be the ordered edge sequence of the $fMST_{G^\delta}$. The value transform $T(\Delta)$ is define by

$$T(\Delta) = \{ ||S^{E_1^\delta}|| - i + 2 \mid w(S_{i-1}^{E_1^\delta}) < \Delta \text{ and } w(S_{i+1}^{E_1^\delta}) \geq \Delta \} \quad (4)$$

in which $w(S_i^{E_1^\delta})$ means the weight of the edge $S_i^{E_1^\delta}$.

If all values in the ordered edge sequence are different, $T(\max\{S^{E_1^\delta}\})$ is equal to 2. Thus, to compute a 2-clustering is necessary to eliminate the edge in $fMST_{G^\delta}$ with the highest weight value. Finally, to identify the video scenes from minimum spanning tree of frames is necessary to delete some edges according to a specified criterium.

The edge deletion operation, when applied to a tree, produces two connected components. Here, each connected component is called *frame cluster*, as defined to follow. The process of edge deletion must be agreed to a specified criterium. In this work, we can use two different one: (i) deletion of largest weight edges; or (ii) deletion of edges with weight greater than or equal to a specified threshold. The former is useful when the number of clusters (defined before) is pre-determined. The second one can be considered when the minimum similarity measure between clusters is specified, and can be

considered a special case of the first one when the weights are different. For example, the deletion in Fig. 2 of all edges with weight greater than or equal to a 49 produces the same result when we eliminate the 2 largest weight edges.

Definition 8 (Frame cluster – $C^{*,k}$). Let $fMST_{G^\delta}$ be a minimum spanning tree of frames. Let $C^{*,k}$ denote the k connected components $C_1^{*,k}, C_2^{*,k}, \dots, C_k^{*,k}$ formed by deleting the $k - 1$ largest edges of $fMST_{G^\delta}$ in which $C_i^{*,k} = (N_i, E_i^\delta)$. $C^{*,k}$ is a k -clustering of max spacing.

The number of clusters, and consequently, the number of video scenes is directly related to the number of edge deletion operations. Also, the process to compute the frame cluster is hierarchical in the sense that the edge deletion divides a cluster into two different groups. However, the saliency of a frame cluster component may depend on its size, since components with a small number of frames may represent noise. For example, a black frame or a flashlight frame are probably very dissimilar of all other frames and consequently, the adjacent edges will be the largest weight edges of the $fMST_{G^\delta}$. The frame cluster produced by edge deletion will present a very small number of frames, and consequently, it could be ignored by our analysis.

3 Video Summarization Using Minimum Spanning Tree

In this work, we propose a new approach to video summarization in which the clustering of video frames are based on minimum spanning tree of frames. In Fig 3 is illustrated our method.

Fig. 3. Methodology for video summarization using minimum spanning tree

Thanks to the minimum spanning tree approach, our method eliminates the pre-processing to compute the number of clusters, and also, eliminates the video segmentation step. Thus, we compute the minimum spanning tree of frames from the frame similarity graph that was computed from the video sequence. Afterwards, we delete edges until stability. The concept of stability is related to two approaches: (i) number of desired clusters; and (ii) frame similarity in a cluster. While the former can be related to compress factor of the video summary, the second approach to stability establishes maximum (dis)similarity into a cluster. In both cases, only a keyframe for each cluster is selected. For now, the keyframe selection is done by the frame in the middle time of the cluster.

Fig. 4. Examples of video summarization applied to video "News.mpg" composed by 5 (five) shots: (a) T(35)-clustering; (b) 3-clustering; (c) T(34)-clustering; and (d) T(25)-clustering

Table 1. Experimental results compared to OpenVideo according to [6]

Method	Total Clips	Category 1 (Same Keyframes)	Category 2 (Fewer Keyframes)	Category 3 (More Keyframes)	Category 4 (Mismatched Keyframes)
AGM1	50	18	3	29	24
AGM2	50	28	3	19	3
Visto [3]	50	14	5	31	50
DT [6]	50	30	8	12	10
VSUMM1 [1]	50	20	3	27	5

To present some examples, in Fig. 4(a), Fig. 4(c) and Fig. 4(d) are illustrated the results of application of our method for $T(35) = 2$, $T(34) = 4$ and $T(25) = 5$. It is important to notice that the values 25, 34 and 35 in Fig. 4 represent three highest weight values in minimum spanning tree of frames, however there are 1, 2 and 1 edges with these weights, respectively. Thus, there exist two different options for computing a 3-clustering. The Fig. 4(b) illustrates a 3-clustering.

4 Experiments

In our experiments, we use some video extracted from the repository *Open Video*¹. To visually compare our approach with others, we consider the dataset used in [3] composed by 50 videos in different genres (documentary, lectures, ephemeral, historical, educational). With respect to a comparative analysis, we consider a similar approach to [6] in which we compare the methods according to the summary size and also, the

¹ www.open-video.org

Fig. 5. Examples of static video summary (Video *Oceanfloor Legacy*, segment 02)

Fig. 6. Examples of static video summary (Video *Senses And Sensitivity*, *Introduct. to Lecture 2*)

number of mismatched frames, both related to OpenVideo summary. It is important to note that we consider the same approaches used in [1].

In order to illustrate the tuning of parameters for the proposed method, we consider two different sets of values (δ for clustering, α for connected component size and ϵ for sampling): (i) $\delta = 25$, $\alpha = 10$, $\epsilon = 1$; and (ii) $\delta = 25$, $\alpha = 10$, $\epsilon = 10$. Here, we consider a global measure in order to compute the similarity measure, we consider the histogram intersection from HSV color space, however for some applications, it is necessary to choose similarity measures that saliently represent the frame content like matching of interested points. To facilitate our description, AGM1 and AGM2 denote

the proposed methods with the set of parameters described before, respectively. Also, to realize a visually comparative analysis, we consider the other methods: VSUMM [1], DT [6] and VISTO [3]. In Table 1 we divide our results in four categories (same keyframes, fewer keyframes, more keyframes, mismatched keyframes).

As can be seen, the VSUMM [1] and our approach (AGM2) present similar results. The Fig. 5 and Fig. 6 illustrate examples of video summaries computed by different methods. It is important to note the summaries are compared to the OpenVideo approach. The Fig. 5 illustrated a video summary with more keyframes for both, however the VSUMM computed a redundant frame. The Fig. 6 illustrated a video summary with more keyframes in which the quality of the summary is the same.

5 Conclusion and Further Works

In this work, we propose a divisive and hierarchical method to compute a static video summary without considering segmentation step. Our method is based on computation of MST of the frames. It is well-known that the algorithm to generate the MST is efficient and fast. The analysis of the video summary quality is subjective, however there is no redundancy and only the most salient scenes are represented. From our approach, it is possible to indicate the degree of desired compression. The proposed method try to eliminate the redundancy and non representative scenes, however the tuning of parameters influences the quality of video summary. We expect, in future work, improve the selection of keyframe and also, to automatically tune the parameters.

Acknowledgments

The authors are grateful to PUC Minas (Pontificia Universidade Católica de Minas Gerais), MIC BH (Microsoft Innovation Center Belo Horizonte), CT-Info/MCT/CNPq (Project 551005/2007-6) and FAPEMIG (Project CEX PPM 126/08) for the financial support of this work.

References

1. de Avila Jr., S.E.F., da Luz Jr., A., de Araújo, A.A., Cord, M.: Vsummm: An approach for automatic video summarization and quantitative evaluation. In: SIBGRAPI 2008: Proceedings of the 2008 XXI Brazilian Symposium on Computer Graphics and Image Processing, pp. 103–110. IEEE Computer Society, Washington (2008)
2. Besiris, D., Fotopoulou, F., Economou, G., Fotopoulos, S.: Video summarization by a graph-theoretic fcm based algorithm. In: 15th International Conference on Systems, Signals and Image Processing, IWSSIP 2008, 25–28, pp. 511–514 (2008)
3. Furini, M., Geraci, F., Montangero, M., Pellegrini, M.: Visto: visual storyboard for web video browsing. In: Proceedings of the 6th ACM International Conference on Image and Video Retrieval, CIVR 2007, pp. 635–642. ACM, New York (2007)
4. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: A review. *ACM Comput. Surv.* 31(3), 264–323 (1999)
5. Kleinberg, J., Tardos, É.: *Algorithm Design*. Addison-Wesley, Reading (2006)

6. Mundur, P., Rao, Y., Yesha, Y.: Keyframe-based video summarization using delaunay clustering. *Int. J. on Digital Libraries* 6(2), 219–232 (2006)
7. Rui, Y., Huang, T.S., Mehrotra, S.: Browsing and retrieving video content in a unified framework. In: 1998 IEEE Second Workshop on Multimedia Signal Processing, 7-9, pp. 9–14 (1998)
8. Tan, Y.P., Lu, H.: Video scene clustering by graph partitioning. *Electronics Letters* 39(11), 841–842 (2003)
9. Yeung, M., Yeo, B.L.: Video visualization for compact presentation and fast browsing of pictorial content. *IEEE Transactions on Circuits and Systems for Video Technology* 7(5), 771–785 (1997)
10. Zahn, C.: Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers* C-20(1), 68–86 (1971)
11. Zhu, S.H., Liu, Y.-C.: Automatic video partition for high-level search. *IJCSSES International Journal of Computer Sciences and Engineering Systems* 2(3), 163–172 (2008)