Static Video Summarization through Optimum-Path Forest Clustering

G.B. Martins[1], L.C.S. Afonso[1], D. Osaku[3],
Jurandy Almeida[2], and J.P. Papa[1,⋆]

[1] Department of Computing, São Paulo State University, UNESP
17033-360, Bauru, SP, Brasil
{gui.bmartins.unesp,sugi.luis}@gmail.com, papa@fc.unesp.br

[2] Institute of Science and Technology, Federal University of São Paulo, UNIFESP
12231-280, São José dos Campos, SP, Brazil
jurandy.almeida@unifesp.br

[3] Department of Computer Science, Federal University of São Carlos, UFSCar
13565-905, São Carlos, SP, Brazil
danosaku@hotmail.com

Abstract. This paper introduces the Optimum-Path Forest (OPF) classifier for static video summarization, being its results comparable to the ones obtained by some state-of-the-art video summarization techniques. The experimental section has been conducted using several image descriptors in two public datasets, followed by an analysis of OPF robustness regarding one ad-hoc parameter. Future works are guided to improve OPF effectiveness on each distinct video category.

Keywords: video summarization, optimum-path forest, clustering.

1 Introduction

Recent advances in technology have increased the availability of video data, creating a strong requirement for efficient systems to manage those materials. Making efficient use of video information requires that data to be accessed in a user-friendly way. For such purpose, it is important to provide to the users a concise video representation to give an idea of a video content, without having to watch it entirely, so that a user can decide whether to watch the entire video or not. This has been the goal of a quickly evolving research area known as video summarization [12,22].

Techniques for video summarization are commonly classified in static or dynamic ones. Static techniques are the main goal of the former methodologies to obtain keyframes of the original video in order to compose the compressed representation, whereas the dynamic techniques aim to find out a collection of segments (set of frames nearby the keyframes) to provide more reasonable summaries, which can also include sound effects [2].

[⋆] The authors are grateful to FAPESP grants #2012/06472-9 and #2013/20387-7, as well as to CNPq grants #303182/2011-3 and #470571/2013-6.

E. Bayro-Corrochano and E. Hancock (Eds.): CIARP 2014, LNCS 8827, pp. 893–900, 2014.

The basic steps for a static video summarization consist in extracting descriptors from each frame and clustering them in the feature space using some unsupervised technique. Then, the most representative sample (frame) of each cluster is used as the keyframe to compose the video summary. One of the most used unsupervised classification technique for this purpose is the well-known k-means, mainly due to its simplicity and reasonable results in several applications. The idea of k-means is to find the samples that fall in the center of each cluster, being an interesting approach to obtain keyframes: it means that a sample in the center of a class tends to better represent it, since it has the mean shortest distance among all samples of that cluster. However, k-means requires a priori knowledge of the number of clusters, which generally obligates a post-processing step, since the number of clusters found by k-means may not be the desired one.

Recently, Rocha et al. [17] presented an unsupervised version of the Optimum-Path Forest (OPF) classifier, which models the task of clustering as a graph partition into optimum-path trees (OPTs) by a competition process between some key samples (prototypes). Therefore, a sample that belongs to a given OPT means it is more strongly connected to the root (prototype) of this tree to any other root in that graph. One interesting skill of OPF is that it finds the number of clusters on-the-fly, i.e., OPF does not require the knowledge about the number of frames of that video, which makes it interesting for automatic video summarization. In addition, the OPF prototypes are encoded by the samples with highest density, which means that such samples tend to be located at the center of the classes, similarly to k-means.

In this paper, we introduce the OPF classifier for static video summarization, and also show it can achieve results comparable to the ones obtained by some state-of-the-art video summarization techniques in two public datasets, but with less user effort. The remainder of the paper is organized as follows: Sections 2 and 3 present the related works and the OPF background theory, respectively. Section 4 discusses the experiments, and Section 5 states conclusions.

2 Related Works

A comprehensive review of video summarization approaches can be found in [12,22]. In this work, we are interested in algorithms that produce a collection of static video frames, also known as storyboard [22]. Mundur et al. [13] presented an approach for video summarization based on the Delaunay Triangulation (DT), which has several phases. At the beginning, a lot of redundant information is discarded by a pre-sampling step and, hence, instead of considering all the video frames, only a subset is taken. Then, the Principal Component Analysis is applied on a matrix formed by color histograms extracted from the remaining frames, reducing its dimensionality. After that, the Delaunay diagram is built. Finally, the clusters are obtained by separating edges in the diagram.

The STIll and MOving Video Storyboard (STIMO) [8], is a summarization technique designed to produce on-the-fly video summaries. Initially, a pre-sampling step is applied in order to discard a lot of redundant information, taking only a subset of video frames. The remaining frames are then converted into color histograms and stored in a feature-frame matrix. Next, similar frames are grouped together by a clustering method based on an improved version of the Furthest-Point-First algorithm. For obtaining the number of clusters, the pairwise dissimilarity of consecutive frames is computed according to Generalized Jaccard Distance. Finally, a post-processing step is performed by removing meaningless frames from the storyboard.

The Video SUMMarization (VSUMM) [5] is a similar approach to cope with the video summarization problem in which the clustering step is achieved by an improved version of the k-means algorithm. For that, the frames are initially grouped in sequential order instead of randomly distributed between the clusters. Thereafter, the frames are grouped by the traditional k-means algorithm. Finally, one frame per cluster is selected for the summary.

Finally, the VIdeo Summarization for ONline applications (VISON) [1] is a summarization technique that operates directly in the compressed domain, allowing the online usage. For each frame of an input sequence, visual features are extracted from the video stream for describing its visual content. After that, a simple and fast algorithm is used to detect groups of video frames with a similar content and also to select a representative frame per each group. Finally, the selected frames are filtered in order to avoid possible redundant or meaningless frames in the video summary.

3 Optimum-Path Forest Clustering

The OPF classifier interprets the dataset set as a graph, whose nodes are the samples and the arcs connect pairs of samples that satisfy a given *adjacency relation.* For a suitable *path-value (connectivity) function*, the optimum-path forest algorithm [7] partitions the graph into optimum-path trees (clusters) rooted at some key samples, named *prototypes.* The prototypes compete among themselves for the most closely connected samples in the dataset, such that each sample is assigned to the tree whose prototype offers to it an optimum path.

Let $\mathcal{Z}$ be a dataset such that for every sample $s \in \mathcal{Z}$ there exists a feature vector $\boldsymbol{v}(s)$. Let $d(s,t)$ be the distance between s and t in the feature space. For instance, $d(s,t) = \|\boldsymbol{v}(t) - \boldsymbol{v}(s)\|$ — the Euclidean distance between $\boldsymbol{v}(t)$ and $\boldsymbol{v}(s)$. A graph $(\mathcal{Z}, \mathcal{A}_k)$ can be defined such that the arcs $(s,t) \in \mathcal{A}$ connect k-nearest neighbors in the feature space. The arcs are weighted by $d(s,t)$ and the nodes $s \in \mathcal{Z}$ are weighted by a probability density value $\rho(s)$:

$$\rho(s) = \frac{1}{\sqrt{2\pi\sigma^2}|\mathcal{A}_k(s)|} \sum_{\forall t \in \mathcal{A}_k(s)} \exp\left(\frac{-d^2(s,t)}{2\sigma^2}\right), \tag{1}$$

where $|\mathcal{A}_k(s)| = k$, $\sigma = \frac{d_f}{3}$, and d_f is the maximum arc weight in $(\mathcal{Z}, \mathcal{A}_k)$. This parameter choice considers all adjacent nodes for density computation, since a Gaussian function covers most samples within $d(s,t) \in [0, 3\sigma]$. Moreover, since $\mathcal{A}_k$ is asymmetric, symmetric arcs must be added to it on the plateaus of the probability density function (pdf) in order to guarantee a single root per maximum. The solution proposed by Rocha *et al.* [17] to find the best value of k, i.e., k^*, considers the minimum graph cut among all clustering results for $k \in [1, k_{\max}]$ ($k_{\min} = 1$), according to the normalized measure suggested by Shi and Malik [18].

The method defines a path π_t as a sequence of adjacent samples starting from a root $R(t)$ and ending at a sample t, being $\pi_t = \langle t \rangle$ a trivial path and $\pi_s \cdot \langle s, t \rangle$ the concatenation of π_s and arc (s, t). It assigns to each path π_t a value $f(\pi_t)$ given by a connectivity function f. A path π_t is considered optimum if $f(\pi_t) \geq f(\tau_t)$ for any other path τ_t.

Among all possible paths π_t from the maxima of the pdf, the method assigns to t a path whose minimum density value along it is maximum. That is, the method finds $V(t) = \max_{\forall \pi_t \in (\mathcal{Z}, \mathcal{A}_k)}\{f(\pi_t)\}$ for $f(\pi_t)$ defined by:

$$
\begin{aligned}
f(\langle t \rangle) &= \begin{cases} \rho(t) & \text{if } t \in \mathcal{R} \\ \rho(t) - \delta & \text{otherwise,} \end{cases} \\
f(\langle \pi_s \cdot \langle s, t \rangle \rangle) &= \min\{f(\pi_s), \rho(t)\}, \qquad (2)
\end{aligned}
$$

for $\delta = \min_{\forall (s,t) \in \mathcal{A}_k | \rho(t) \neq \rho(s)} |\rho(t) - \rho(s)|$ and $\mathcal{R}$ being a root set, discovered on-the-fly, with one element per each maximum of the pdf. It should be noted that higher values of δ reduce the number of maxima. We are setting $\delta = 1.0$ and scaling real numbers $\rho(t) \in [1, 1000]$ in this work. The OPF algorithm maximizes the connectivity map $V(t)$ by computing an optimum-path forest — a predecessor map P with no cycles that assigns to each sample $t \notin \mathcal{R}$ its predecessor $P(t)$ in the optimum path from $\mathcal{R}$ or a marker *nil* when $t \in \mathcal{R}$.

4 Experiments

In this section, we describe the methodology used to assess the robustness of OPF clustering in the context of static video summarization, as well as the experimental results. The proposed OPF-based video summarization can be divided in three steps: (i) video sampling, (ii) feature extraction, and (iii) video summarization, as depicted in Figure 1.

The first step uses a pre-sampling approach for extracting frames from the videos to be summarized. The *video sampling* was performed by the well-known *ffmpeg* tool[1] in a sampling rate of one frame per second in two public datasets[2]: Open Video and YouTube. The former contains 50 videos randomly selected from the Open Video Project[3], which are distributed among three different genres (i.e., documentary, educational, and lecture) and their duration varies from 1 to 4 minutes. The latter is composed of 40 videos collected from the YouTube[4], which are distributed among five genres (i.e., sports, news, tv-shows, commercials, and home videos) and their duration varies from 1 to 10 minutes.

[1] `http://www.ffmpeg.org/` (As of August 2014).

[2] `http://sites.google.com/site/vsummsite/` (As of August 2014).

[3] `http://www.open-video.org/` (As of August 2014).

[4] `http://www.youtube.com/` (As of August 2014).

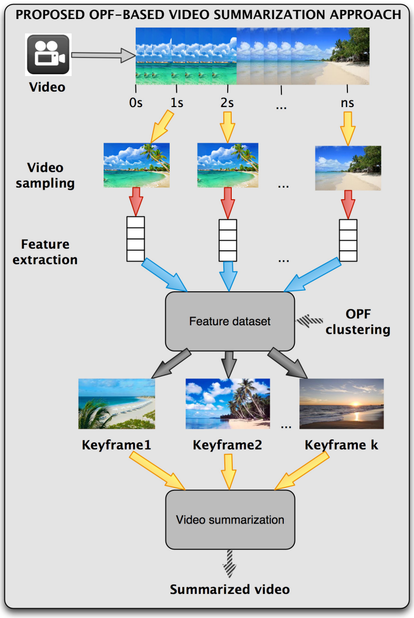


Fig. 1. Steps performed during video summarization

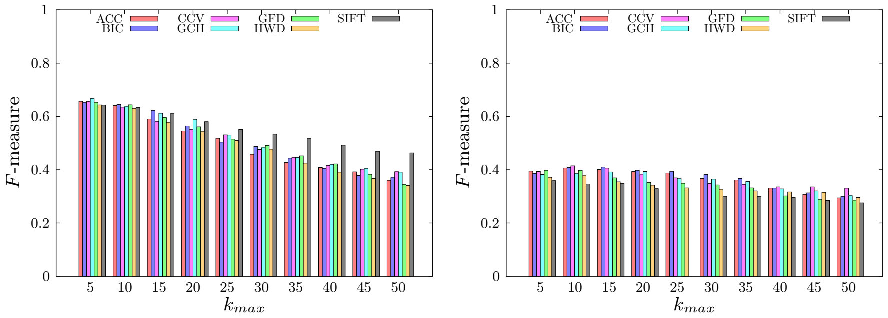
The second phase performs the *feature extraction* from each frame extracted in the previous step. We have evaluated six different descriptors for such task: Auto Color Correlogram (ACC) [9], Color Coherent Vector (CCV) [14], Border/Interior pixel Classification (BIC) [20], and Global Color Histogram (GCH) [21], for encoding color information; Generic Fourier Descriptor (GFD) [23] and Haar-Wavelet Descriptor (HWD) [10], for analyzing spectral properties. For more details regarding those image descriptors, refer to [15]. In addition, we built a Bag-of-Features (BoF) representation [19] using SIFT (Scale-Invariant Feature Transform) features [11]. For that, we constructed a visual dictionary using k-Means with $k = 4000$ visual words (k has been empirically chosen). Therefore, after the feature extraction step, we have seven feature-based datasets for each video dataset.

The final step concerns with the *video summarization* itself. For comparison, we used the results reported by four recently proposed static summarization techniques. In the Open Video dataset, we compared OPF clustering against DT [13], STIMO [8], VSUMM [5], and VISON [1]. Additionally, the summaries produced by OPF were compared with the storyboards presented at the Open Video (OV), which are generated using the algorithm of DeMenthon et al.[6] and added to some manual intervention to refine the results. On the other hand, in the YouTube dataset, OPF was compared against VSUMM and VISON[5].

[5] Notice the OPF results for all videos can be checked out at `http://www.liv.ic.unicamp.br/~jurandy/opfvs/` (As of August 2014).

In this work, we adopted a subjective evaluation method to assess the quality of video summaries, known as *Comparison of User Summaries* (CUS) [5]: initially, the subjects are asked to watch the whole video, and further they are oriented to select a subset of frames that is able to summarize the video content. Each subject is free to select any number of frames to compose his/her summaries. Finally, their summaries are compared with the summaries provided by the algorithms.

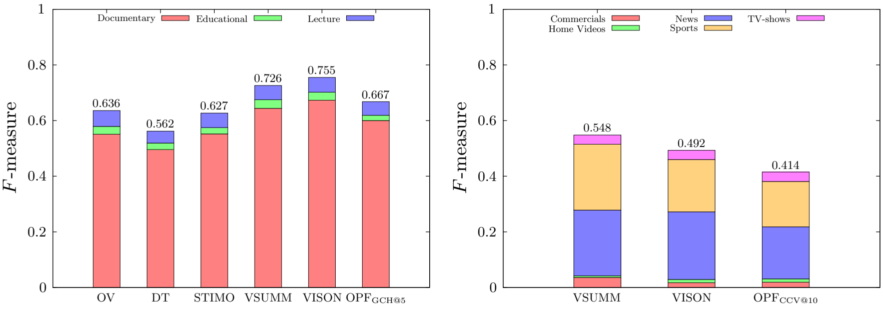
In order to compare frames from different summaries, we used the pixel-wise matching method proposed by Almeida et al. [3]. Once two frames are matched, they are removed from the next iteration of the comparing procedure. Thus, the comparison between the user summary and the automatic summary is led to the number of frames gathered. The standard measures *precision* and *recall* can then be used to evaluate the automatic summary, being precision the ratio of the number of matching frames to the total number of frames in the automatic summary. Recall is the ratio of the number of matching frames to the total number of frames in the user summary. In this paper, we choose the F-measure as the metric used for evaluating the performance, mainly due to the trade-off between *precision* and *recall.* The increase of one value decreases the second, and vice-versa. In addition, it is important to shed light over that F-measure is one of the most used approaches for the analysis of video summaries.



(a) Open Video (b) YouTube

Fig. 2. The mean F-measure achieved by each of the descriptors for different values of k_{max}

As aforementioned, OPF computes clusters on-the-fly based on optimum paths and a variable k_{max} (Section 3), which defines the maximum number of nearest neighbors to be considered during cluster computation. Although the reader may argue that the algorithm does not compute clusters fully automatically, it is important to highlight that changing the value of k_{max} causes less impact on the final result than varying the value of k for k-means, for instance. For each feature-based dataset, we evaluated k_{max} value within the range $[5, 50]$ with steps of 5. Finally, we choose the value of k_{max} that maximizes F-measure. Figure 2 shows the F-measure values for OPF with different values of k_{max}, in which GCH descriptor with $k_{max} = 5$ and CCV descriptor with $k_{max} = 10$ have been the combination that maximized the F-measure for OPF for Open Video and YouTube datasets, respectively.



(a) Open Video

(b) YouTube

Fig. 3. The mean F-measure achieved by different approaches for each the video category

From Figure 3, the user can observe OPF has been the third best approach for Open Video dataset (Figure 3a), as well as has been slightly less accurate than VISION approach in YouTube dataset (Figure 3b). It is worth noting to stress that OPF requires less user interaction than other methods: VISON has some user parameters, and VSUMM requires the knowledge of parameter k of k-means, for instance. Although the reader may argue OPF has the parameter k_{max} to be set, it is much more intuitive than k-means parameter, and also it is less prone to errors, since there are some situations in which variations on k_{max} may not affect the results a lot, as stated in YouTube dataset (Figure 3b).

5 Conclusions

In this paper, we have introduced OPF clustering for automatic static video summarization, being its effectiveness comparable to the ones obtained by some state-of-the-art video summarization techniques in two public datasets. Several image descriptors have been employed to assess the suitability of OPF for video summarization tasks. Currently, we are working on improving OPF F-measure on each video category separately, aiming to outperform VISON and VSUMM. Future work includes the extension of our framework to consider learning-to-rank methods (e.g., genetic programming [4]) for combining descriptors and hierarchical clustering methods [16].

References

1. Almeida, J., Leite, N.J., Torres, R.S.: VISON: VIdeo Summarization for ONline applications. Pattern Recognition Letters 33(4), 397–409 (2012)
2. Almeida, J., Leite, N.J., Torres, R.S.: Online video summarization on compressed domain. Journal of Visual Communication and Image Representation 24(6), 729–738 (2013)
3. Almeida, J., Torres, R.S., Leite, N.J.: Rapid video summarization on compressed video. In: IEEE Int. Symp. Multimedia (ISM 2010), pp. 113–120 (2010)

4. Andrade, F.S.P., Almeida, J., Pedrini, H., da Torres, R.S.: Fusion of local and global descriptors for content-based image and video retrieval. In: Alvarez, L., Mejail, M., Gomez, L., Jacobo, J. (eds.) CIARP 2012. LNCS, vol. 7441, pp. 845–853. Springer, Heidelberg (2012)
5. Avila, S.E.F., Lopes, A.P.B., Luz Jr., A., Araújo, A.A.: VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method. Pattern Recognition Letters 32(1), 56–68 (2011)
6. DeMenthon, D., Kobla, V., Doermann, D.S.: Video summarization by curve simplification. In: ACM Int. Conf. Multimedia (MM 2008), pp. 211–218 (1998)
7. Falcão, A., Stolfi, J., Lotufo, R.: The image foresting transform theory, algorithms, and applications. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(1), 19–29 (2004)
8. Furini, M., Geraci, F., Montangero, M., Pellegrini, M.: STIMO: STIll and MOving video storyboard for the web scenario. Multimedia Tools Appl. 46(1), 47–69 (2010)
9. Huang, J., Kumar, R., Mitra, M., Zhu, W.J., Zabih, R.: Image indexing using color correlograms. In: IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR 1997), pp. 762–768 (1997)
10. Jacobs, C.E., Finkelstein, A., Salesin, D.: Fast multiresolution image querying. In: Int. Conf. Computer Graphics and Interactive Techniques (SIGGRAPH 1995), pp. 277–286 (1995)
11. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Computer Vision 60(2), 91–110 (2004)
12. Money, A.G., Agius, H.W.: Video summarization: A conceptual framework and survey of the state of the art. J. Visual Communication and Image Representation 19(2), 121–143 (2008)
13. Mundur, P., Rao, Y., Yesha, Y.: Keyframe-based video summarization using Delaunay clustering. Int. J. on Digital Libraries 6(2), 219–232 (2006)
14. Pass, G., Zabih, R., Miller, J.: Comparing images using color coherence vectors. In: ACM Int. Conf. Multimedia (ACM-MM 1996), pp. 65–73 (1996)
15. Penatti, O.A.B., Valle, E., Torres, R.S.: Comparative study of global color and texture descriptors for web image retrieval. Journal of Visual Communication and Image Representation 23(2), 359–380 (2012)
16. Rocha, A., Almeida, J., Nascimento, M.A., Torres, R., Goldenstein, S.: Efficient and flexible cluster-and-search for CBIR. In: Blanc-Talon, J., Bourennane, S., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2008. LNCS, vol. 5259, pp. 77–88. Springer, Heidelberg (2008)
17. Rocha, L.M., Cappabianco, F.A.M., Falcão, A.X.: Data clustering as an optimum-path forest problem with applications in image analysis. International Journal of Imaging Systems and Technology 19(2), 50–68 (2009)
18. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8), 888–905 (2000)
19. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: IEEE Int. Conf. Computer Vision (ICCV 2003), pp. 1470–1477 (2003)
20. Stehling, R.O., Nascimento, M.A., Falcão, A.X.: A compact and efficient image retrieval approach based on border/interior pixel classification. In: ACM Int. Conf. Information and Knowledge Management (CIKM 2002), pp. 102–109 (2002)
21. Swain, M.J., Ballard, B.H.: Color indexing. Int. J. Computer Vision 7(1), 11–32 (1991)
22. Truong, B.T., Venkatesh, S.: Video abstraction: A systematic review and classification. ACM Trans. Multimedia Comput. Commun. Appl. 3(1), 1–37 (2007)
23. Zhang, D., Lu, G.: Shape-based image retrieval using generic fourier descriptor. Signal Processing: Image Communication 17(10), 825–848 (2002)