Automatic Video Summarization Using the Optimum-Path Forest Unsupervised Classifier

César Castelo-Fernández(✉) and Guillermo Calderón-Ruiz

School of Systems Engineering, Santa María Catholic University, Arequipa, Peru
{ccastelo,gcalderon}@ucsm.edu.pe

Abstract. In this paper a novel method for video summarization is presented, which uses a color-based feature extraction technique and a graph-based clustering technique. One major advantage of this method is that it is parameter-free, that is, we do not need to define neither the number of shots or a consecutive-frames dissimilarity threshold. The results have shown that the method is both effective and efficient in processing videos containing several thousands of frames, obtaining very meaningful summaries in a quick way.

Keywords: Optimum-path forest classifier · Video summarization · Shot detection · Clustering · Video processing

1 Introduction

Nowadays, huge amounts of multimedia information exist thanks to the popularization of smart portable devices like cell phones or tablets, which have cameras capable of recording high quality pictures and videos.

Such volume of information makes it necessary to have software capable of summarizing this information, allowing us to store only the most important parts of the videos. For example, 24-hour surveillance videos, that need to be analyzed daily, could be summarized into few-minutes clips. This problem could be solved using video processing and computer vision techniques.

To perform the analysis of the video's content, we need to extract features from each frame of it, more specifically, we need to characterize the color, texture and shape of the images composing the video. Then, the video is splitted into scenes and shots, which are the parts that compose it. This process is known as shot detection and it is the most important part of this kind of systems.

On the other hand, traditional approaches for shot detection consider the analysis of the level of dissimilarity between consecutive frames, setting a new shot when the dissimilarity is higher than a certain threshold. The problem with this approach is determining a suitable threshold for each video.

This paper presents a new approach for automatic video summarization, based on the application of the Optimun-Path Forest unsupervised classifier, a graph-based clustering technique known for being both fast and accurate.

A. Pardo and J. Kittler (Eds.): CIARP 2015, LNCS 9423, pp. 760–767, 2015.
DOI: 10.1007/978-3-319-25751-8_91

Moreover, this new approach does not need to know a priori the number of shots and scenes as other techniques do.

The rest of the document is organized as follows. Section 2 presents the main concepts of video summarization systems, Section 3 explains the techniques used for the proposed method, while Sections 4 and 5 present the evaluation and the conclusions, respectively.

2 Video Summarization

Formally speaking, a video can be defined as a set $V = \langle f_1, f_2, \ldots, f_n \rangle$ of frames f_i, that are images of M x N pixels. A video is made of the union of shots, which are separated by transitions. A transition T_i between the shots S_i and S_{i+1} can be represented by the pair $(s, t), s < t$, which are the indexes of the frames that form the transition, such that, $S_i = \langle \ldots, f_{s-1}, f_s \rangle$ and $S_{i+1} = \langle f_t, f_{t+1}, \ldots \rangle$. These transitions can be abrupt, when $t = s+1$ (i.e. one shot starts immediately after the other) or gradual, when $t > s + 1$ (i.e. the two shots are superposed), being that the latter have edition effects, like fades or dissolutions.

In addition, the set of shots is grouped into scenes, which could be seen as a set of related shots developed in the same environment, also known as shots taken by the same camera.

The work developed by Chen et al. [2] presents an algorithm to detect the transitions in a video by using a threshold for the distance (dissimilarity measure) between two consecutive frames. This algorithm is very fast but it is very difficult to choose a suitable threshold, because it can be very different among videos.

Jadhav and Jadhav [4] developed a video summarization method that uses higher order color moments as the feature extraction technique. This method aims at the detection of shot boundaries by computing different statistics with the frames. However, it needs several thresholds for the shot boundary detection.

The method proposed by Ejaz et al. [3] uses an adaptive correlation scheme of color-based features which aims at the detection of key frames by using a threshold for the correlation level among them. The main drawback of this method is choosing a suitable threshold for the correlation levels.

Ren et al. [6] proposed a fuzzy approach that is able to classify the frames in the video according to the transitions present between them and according to camera motion analysis. The main idea is to generate a summary of the video according to the activities being performed inside it. One drawback is the use of a threshold to determine the final size of the summary generated.

Finally, Zhou et al. [9] proposed a video summarization method based on the use of a fuzzy c-means algorithm together with audio-visual features extracted from the video. The main problem of this technique is determining a suitable value for c (number of shots).

3 Automatic Video Summarization by Optimum-Path Forest

The method proposed in this paper relies on the use of a clustering algorithm to divide the video into shots.

We let the frames form groups in a natural way, that is, every group contains the most similar frames to each other. The reason for this is that a video is nothing more than a sequence of very similar pictures placed one next to the other at a certain frame rate. However, this is true for portions of the video (shots), not for the entire video, i.e., all the frames inside a shot are very similar to each other and they are different to frames in other shots.

This is the reason for using clustering. A clustering algorithm tries to find an optimal partition inside a group of objects, following the rule that every group has the highest level of similarity among their elements and the lowest level of similarity to the objects in the other clusters.

To perform this task, we need the objects to be represented numerically, more specifically as a feature vector, that it, a set of N numbers that represent the object as a point in a N-dimensional space. It is able to dictate whether two given images are similar or dissimilar. For the case of videos, they are obtained by applying one or more image processing techniques (i.e. feature extraction), which aim at the representation of the color, texture and shape of the frames.

Among the feature extraction techniques, we could mention color histograms, Gabor filters or Fourier descriptors, which are focused on representing color, texture and shape, respectively. Then, some distance function is needed to compute the level of dissimilarity between two vectors (e.g. Euclidean distance or Mahalanobis distance). Moreover, a very interesting option, and the one chosen for this work, is *Border-Interior Pixel Classification (BIC)*. This technique was chosen because it is both accurate for representing the color and fast enough to process the frames in the videos.

Regarding the clustering techniques, the most used approaches in the literature are the *k-means* algorithm and its variants *fuzzy k-means* or *k-medians*, which are all partition-based approaches. Also, we can mention probability-based approaches like *Mean-Shift*, *DBSCAN* and *Expectation-Maximization*. And, between others, we can also talk about graph-based approaches, like the ones based on the *Minimum-Spanning Tree (MST)* or the *Optimum-Path Forest (OPF)*, which is the one chosen for this work. We chose the OPF algorithm, because it is a threshold-free approach and it is both accurate and fast enough to find the shots in the video in a very short amount of time.

3.1 Feature Extraction Through Border/Interior Pixel Classification

Border-Interior Pixel Classification (BIC) [8] is a color-based technique proposed to address the well-known issue of global color representation presented by the traditional color histograms. This problem raises when we have two very different images with a very similar color distribution, i.e., two very different images will have very similar feature vectors.

As a solution, BIC performs a previous step of pixel classification into two possible kind of pixels, border and interior. For this, a quantization process has to be performed, transforming the color range of the pixels from, say, 256 gray levels (original image) to, say, 4 or 8 gray levels. Thus, an interior pixel is the one that is surrounded by pixels of the same (quantized) color and a border pixel is the one that is surrounded by at least one different color pixel. In other words, the interior pixels are the homogeneous regions of the image and the border pixels are the edges surrounding them. Then, separate histograms are computed for both interior and border pixels and the union of them is the final vector.

Furthermore, an improved distance measure is used instead of the traditional euclidean distance. This distance is called *dLog* and uses a logarithmic scale instead of a decimal one, aiming at the dissipation of the effect of very large values in the histograms (e.g. the image's background). It is defined by the Equation 3.1.

$$dLog(q,d) = \sum_{i=0}^{i<M} |f(q[i]) - f(d[i])| \tag{3.1}$$

where M is the dimension of the vectors q and d and f is defined as:

$$f(n) = \begin{cases} 0 & \text{if } x = 0 \\ 1 & \text{if } 0 < x \leq 1 \\ log_2 x + 1 & \text{otherwise} \end{cases} \tag{3.2}$$

3.2 Clustering Through the Optimum-Path Forest Unsupervised Classifier

The *Optimum-Path Forest (OPF)* [7] algorithm's main goal is to find an optimal partition of the training set with the help of a graph, which is used to represent the samples being grouped. For this, every sample is mapped to a node in the graph, representing it as a feature vector. The optimal partition OPF seeks for is represented as a forest of optimal trees rooted in specially-chosen samples called *prototypes*. A cluster is made of one or more trees.

OPF uses a searching algorithm which aims at finding an optimum path for every sample in the training set, which starts in one of the chosen prototypes. In this sense, this algorithm could be seen as a multiple-source and multiple-end shortest-path algorithm, defined by Equation 3.3.

$$\mathcal{V}(t) = \max_{\forall \pi_t^s \in (\mathcal{N}, \mathcal{A})} \{f(\pi_t^s)\} \tag{3.3}$$

where π_t^s is the path from the node $s \in \mathcal{S}$ (set of prototypes) to the node t, $(\mathcal{N}, \mathcal{A})$ is the set of nodes (as defined by the adjacency relationship $\mathcal{A}$) and f is a function that measures the cost of the path π_t^s.

The set of prototypes $\mathcal{S}$ is chosen by computing a *Probability Density Function (PDF)* over the training set and then finding the maxima of this PDF. However, this PDF does not use the traditional (radial) Gaussian function to

weight the relationship between every sample and its neighbors. It uses a modified version of the Gaussian function that uses the k nearest neighbors to every sample s instead of considering all the neighbors that are inside a certain radius. This approach has the advantage of better dealing with arbitrary-shape clusters by not assuming that all clusters has a circular shape, but allowing them to be represented in its natural way. Equation 3.4 dictates how the PDF is computed.

$$\rho(s) = \frac{1}{\sqrt{2\pi\sigma^2}|\mathcal{A}(s)|} \sum_{t \in \mathcal{A}(s)} \exp\left(\frac{-d^2(s,t)}{2\sigma^2}\right) \tag{3.4}$$

which computes a gaussian function centered on s where $\mathcal{A}$ is the adjacency function defined by k and σ is defined as: $\sigma = \max_{\forall(s,t)\in A}\left\{\frac{d(s,t)}{3}\right\}$.

While we need to enter the number k of neighbours used to create the k-nn graph, it can be computed automatically by measuring the quality of the graph cut and then optimizing this value for $[1, k_{max}]$, as proposed by Rocha et al. [7]. Finally, after finding the PDF maxima, we need to reduce the number of clusters found by using β to avoid very small clusters, as explained by Rocha et al. [7].

3.3 Shot and Scene Detection Through Two-Level Clustering

The clustering process explained above is performed using the feature vectors obtained from the video's frames. The result is a set of groups which correspond to the shots of the video. Then, we would like to choose one frame for each shot that better represents it. This frame is called the key-frame and corresponds to the centroid of the cluster.

Furthermore, after computing all the key-frames, we use them to perform the last process of our method, namely, the scene detection process. Based on the same principles behind the use of clustering to find the shots, we perform a new clustering process using only the key-frames. As a result, the groups found in this task will correspond to the scenes of the video.

Finally, we also compute the scene key-frames for this clusters and use them to generate the summary of the video. For this, we extract a small set of frames before and after each scene key-frame and put them together according to the time position of each key-frame. According to the job developed by Pfeiffer et al. [5], the minimum size a scene should have is 3.25 seconds in order to be analyzed properly by a person.

4 Evaluation

The validation process of the proposed method was initially made using commercial videos of different sizes. Table 1 summarizes the characteristics of the videos. A manual inspection process was performed to obtain the shot detection ground-truth for each video.

Regarding the statistical approach used to compare our method with other methods, we used the ROC curve, the most used method to compare information

Table 1. Videos used for the tests.

Video name	Resolution	Length	Frames	FPS	Shots
Taboo	480x320	04:52 min.	7008	24	201
Say it right	480x320	03:56 min.	5664	24	216
Crazy Frog	352x288	03:20 min.	5006	25	82
Bendita tu luz	480x320	04:10 min.	6007	24	146
Destino de fuego	480x360	03:59 min.	7191	30	175

retrieval algorithms. A ROC curve allows us to evaluate the global behavior of a technique, i.e., it is not biased by a particular choice of parameters.

We chose the threshold-based method proposed by Chen et al. [2] and also we chose a similar work, developed by Castelo [1], where a k-means algorithm is used to perform the clustering process. Furthermore, a threshold-based method is used in this work to determine the value of k.

Figure 1 presents the ROC curves obtained for the three methods, our OPF-based method, the threshold-based method and the k-means-based method.

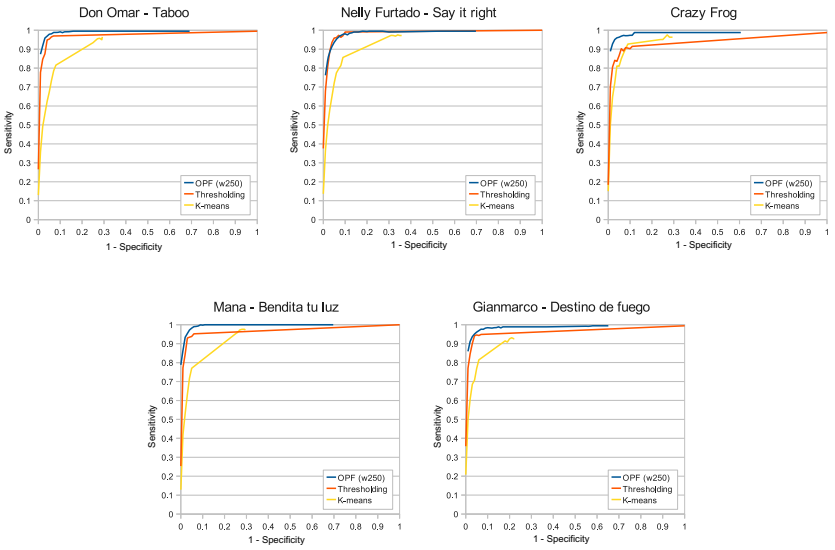

Fig. 1. ROC curves for the three methods, using the test videos. Here: $1 \leq k \leq 20, 0 \leq \beta \leq 500$ (OPF) y $0 \leq \mu \leq 0.99, 0 \leq \delta \leq 0.99$ (Thresholding and k-means).

Table 2 summarizes the ROC curves computed and presents, for each technique, the combination of parameters that lead to the better results according to the ROC curve.

We used different threshold values for both methods to create the ROC curve (thresholds μ and δ are used to detect abrupt and gradual transitions separately)

Table 2. False Positive Rate, True Positive Rate, Accuracy and Execution Time (seconds) for the three methods.

Video	Method	k/μ	β/δ	FPR	TPR	Acc.	Exec.Time
Taboo	OPF	7	160	**0.04**	**0.99**	**0.97**	0.564 ± 0.077
	Thresh.	0.00	0.04	**0.04**	0.97	0.96	**0.001 ± 0.001**
	k-means	0.05	0.05	0.08	0.82	0.87	1.193 ± 2.048
Say it right	OPF	6	40	0.07	0.98	**0.96**	0.438 ± 0.062
	Thresh.	0.04	0.12	**0.06**	**0.99**	**0.96**	**0.001 ± 0.001**
	k-means	0.09	0.05	0.09	0.86	0.88	1.184 ± 1.619
Crazy Frog	OPF	14	340	**0.03**	**0.96**	**0.97**	0.396 ± 0.057
	Thresh.	0.04	0.20	0.04	0.89	0.93	**0.001 ± 0.001**
	k-means	0.09	0.05	0.09	0.93	0.92	0.495 ± 1.021
Bendita tu luz	OPF	15	100	**0.02**	**0.97**	**0.97**	0.463 ± 0.065
	Thresh.	0.64	0.04	0.04	0.95	0.95	**0.001 ± 0.001**
	k-means	0.09	0.05	0.06	0.79	0.87	0.732 ± 1.556
Destino de fuego	OPF	14	20	**0.03**	**0.95**	**0.96**	0.605 ± 0.085
	Thresh.	0.00	0.04	0.04	0.94	0.95	**0.001 ± 0.001**
	k-means	0.05	0.05	0.06	0.82	0.88	1.078 ± 1.708

and, for our method, we used different values for the parameters of the OPF algorithm (the number k of nearest neighbors for the k-nn graph and the threshold β, used to automatically find the number of prototypes).

As we can see in Figure 1, the proposed method obtained very good results, i.e., its ROC curve is closer to the top left corner than the other methods'.

Regarding the processing time (Table 2), the proposed method obtained very fast results, considering the number of frames of each video. In comparison with the other methods, as expected, the threshold-based method is faster, since it only performs a lineal analysis of the frames. However, considering the length of the videos we can say that the difference is not very significant. On the other hand, comparing our method with the k-means-based method, it performed so much better regarding both, processing time and accuracy.

Furthermore, to demonstrate the efficiency of the proposed method, it was used with longer videos (movies). Table 3 shows the characteristics of the movies used and the processing times for them. As we can see, the processing time for the movies are very low, considering the number of frames processed. The movie "Ghost Rider" (158572 frames), for example, is processed in 23.85 seconds.

Table 3. Movies used for the time tests with their processing times.

Video	Resolution	Length	Frames	FPS	Processing Times
Batman Forever	320x240	01:56:34 h.	174869	25	27.99
Ghost Rider	320x240	01:50:13 h.	158572	25	23.85
Starship Troopers	320x240	02:08:28 h.	192712	25	36.06

As future work, we will compare the proposed method to a wider range of clustering algorithms.

5 Conclusions

In this work, a novel method for video summarization was presented. This method is based on the use of clustering techniques to find groups inside the video, which correspond to the shots and scenes. As was shown, the method is better than the two methods used for comparison. BIC, the feature extraction technique chosen, has proved to be both effective and efficient since it helped us to obtain low processing times with good accuracy. Overall, the proposed method is very effective to summarize videos, having obtained the best results looking at the ROC curves. Regarding the processing time needed to perform the analysis, our method is very efficient, considering the number of frames in the videos. It only needed 24 seconds to process a video with more than 158000 frames. Furthermore, one strong point of our method is that it does not need to know a priori the number of groups to perform the cluster analysis.

References

1. Castelo-Fernández, C.: Content-based video retrieval through wavelets and clustering. In: Proceedings of the IV Workshop de Visão Computacional, Bauru, São Paulo, Brasil. UNESP (2008)
2. Chen, L.H., Su, C.W., Mark Liao, H.Y., Shih, C.C.: On the preview of digital movies. Journal of Visual Communication and Image Representation (2003)
3. Ejaz, N.: Tayyab Bin Tariq, and Sung Wook Baik. Adaptive key frame extraction for video summarization using an aggregation mechanism. Journal of Visual Communication and Image Representation **23**(7), 1031–1040 (2012)
4. Jadhav, P.S., Jadhav, D.S.: Video summarization using higher order color moments (vsuhcm). Procedia Computer Science **45**, 275–281 (2015). International Conference on Advanced Computing Technologies and Applications (ICACTA)
5. Pfeiffer, S., Lienhart, R., Fischer, S., Effelsberg, W.: Abstracting digital movies automatically. Journal of Visual Communication and Image Representation **7**(4), 345–353 (1996)
6. Ren, J., Jiang, J., Feng, Y.: Activity-driven content adaptation for effective video summarization. Journal of Visual Communication and Image Representation **21**(8), 930–938 (2010). Large-Scale Image and Video Search: Challenges, Technologies, and Trends
7. Rocha, L.M., Cappabianco, F.A.M., Falcão, A.X.: Data clustering as an optimum-path forest problem with applications in image analysis. Int. J. Imaging Syst. Technol. **19**, 50–68 (2009)
8. Stehling, R.O., Nascimento, M.A., Falcão, A.X.: A compact and efficient image retrieval approach based on border/interior pixel classification. In: Proceedings of the Eleventh International Conference on Information and Knowledge Management, CIKM '02, pp. 102–109 (2002)
9. Zhou, H., Sadka, A.H., Swash, M.R., Azizi, J., Sadiq, U.A.: Feature extraction and clustering for dynamic video summarisation. Neurocomputing **73**(10–12), 1718–1729 (2010). Subspace Learning/Selected papers from the European Symposium on Time Series Prediction