Temporal- and Spatial-Driven Video Summarization Using Optimum-Path Forest

Guilherme B. Martins, João P. Papa
Department of Computing
São Paulo State University – UNESP
Bauru - SP, Brazil
papa@fc.unesp.br
guilherme-bm@outlook.com

Jurandy Almeida
Institute of Science and Technology
Federal University of São Paulo – UNIFESP
São José dos Campos - SP, Brazil
jurandy.almeida@unifesp.br

***Abstract*—Video summarization aims at generating reduced representations for fast and effective video retrieval and classification. In this paper, we cope with such problem by proposing a temporal- and spatial-driven approach that makes use of the Optimum-Path Forest (OPF) clustering to automatic find the number of keyframes, as well as to extract them to compose the final summary. The experiments in two public datasets show OPF can outperform very recent results, thus achieving a performance comparable to some state-of-the-art techniques.**

***Keywords*-Optimum-Path Forest, Video Summarization**

I. Introduction

Recent advances in image and video technology have allowed users to generate even more high-quality data daily. As a consequence, one also needs efficient and effective mechanisms to store and further retrieve all these digital data. The problem gets worse when we deal with videos, which require much more storage and processing time. In this scenario, it is imperative to have a concise video representation to give an idea of a video content, thus a user can decide whether to watch a entire video or not, without necessarily having to watch it entirely. This has been the goal of a quickly evolving research area known as video summarization [1].

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 at finding out a collection of segments (set of frames nearby the keyframes) to provide more reasonable summaries, which can also include sound effects [2].

A considerable number of works that deal with video summarization can be referred in the literature, being most of them machine learning-oriented. The reason is that video summarization aims at extracting features from frames, for further clustering them in order to group frames with similar content. After that, the most representative sample from each cluster is then elected as the *keyframe*, i.e., the one that shall compose the final video summary.

Almeida et al. [3], for instance, proposed the VISON, which works on compressed videos to allow a fast and effective design of video summaries. Avila et al. [4] presented VSUMM, a video summarization approach based on color information and k-means, which works well in several public datasets. Choi and Kim [5] employed Support Vector Machines for the very same purpose, and Papadopoulos et al. [6] applied a Self-Organized Neural Gas network to produce video summaries, which is able to compute dynamically the number of clusters.

Some years ago, Rocha et al. [7] proposed the Optimum-Path Forest clustering, a graph-based approach that rules a competition process among some key samples in order to conquer the remaining nodes using optimum-path costs. According to some predefined adjacency relation, OPF partitions the dataset into optimum-path trees (clusters) rooted at prototype nodes. This method is easy to use and it has one parameter only, being also able to find the automatic number of clusters on-the-fly.

Very recently, OPF was used to static video summarization with promising results [8], [9]. However, the work by Martins et al. [8] did not consider temporal information during the clustering process to generate the keyframes, thus removing important information from the video summaries. In this work, we propose to cope with this problem by presenting a new approach that allows OPF to consider both temporal and spatial information, thus leading to more accurate video summaries. We show the proposed approach can outperform the previous work based OPF in two public datasets, as well as we can obtain results very close (or even better) to some state-of-the-art techniques to compose video summaries.

The remainder of the paper is organized as follows: Sections II and III present the OPF background theory and the proposed approach, respectively. Section IV discusses the methodology and experiments, and Section V states conclusions.

II. Optimum-Path Forest Clustering

Let $\mathcal{N}$ be a dataset such that for every sample $s \in \mathcal{N}$ there is a feature vector $\vec{v}(s)$. Let $d(s,t)$ be the distance between s and t in the feature space (e.g., $d(s,t) = \|\vec{v}(t) - \vec{v}(s)\|$). The fundamental problem in data clustering is to identify natural groups in $\mathcal{N}$.

A graph $(\mathcal{N}, \mathcal{A})$ is 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 $s \in \mathcal{N}$ are weighted by a density value $\rho(s)$, given by:

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

where $|\mathcal{A}(s)| = k$, $\sigma = \frac{d_f}{3}$, and d_f is the maximum arc weight in $(\mathcal{N}, \mathcal{A})$. This parameter choice considers all nodes for density computation, since a Gaussian function covers most samples within $d(s,t) \in [0, 3\sigma]$. The traditional method to estimate a probability density function (pdf) is by Parzen-window. Equation (1) can provide a Parzen-window estimation based on isotropic Gaussian kernel when we define the arcs by $(s,t) \in \mathcal{A}$ if $d(s,t) \leq d_f$. This choice, however, presents problems with the differences in scale and sample concentration. Solutions for this problem lead to adaptive choices of d_f depending on the region of the feature space [10]. By taking into account the k-nearest neighbors, we are handling different concentrations and reducing the scale problem to the one of finding the best value of k within $[1, k_{\max}]$, for $1 \leq k_{\max} \leq |\mathcal{N}|$. The solution provided by Rocha et al. [7] considers the minimum graph cut provided by the clustering results for $k \in [1, k_{\max}]$, according to a measure suggested by Shi and Malik based on graph cuts [11].

Let a path π_t be 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). Among all possible paths π_t with roots on the maxima of the pdf, we wish to find a path with the lowest density value along it is maximum. Each maximum should then define an influence zone (cluster) by selecting the samples that are more strongly connected to it, according to this definition, than to any other maximum. More formally, we wish to maximize $f(\pi_t)$ for all $t \in \mathcal{N}$ where

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

for $\delta = \min_{\forall (s,t) \in \mathcal{A} | \rho(t) \neq \rho(s)} |\rho(t) - \rho(s)|$ and $\mathcal{R}$ being a root set with one element for each maximum of the pdf. Higher values of delta 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 $f(\pi_t)$ such that the optimum paths form 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 $t \in \mathcal{R}$. In essence, each maximum of the pdf, i.e., prototype, will be the root of an optimum-path tree - OPT (cluster), and the collection of all OPTs originates the optimum-path forest that gives the name to the classifier.

III. Proposed Approach

In this section, we describe the proposed approach based on OPF to obtain static video summaries, which can be divided in six steps: (*i*) video sampling, (*ii*) feature extraction, (*iii*) removal of meaningless frames, (*iv*) clustering, (*v*) removal of redundant keyframes, and (*vi*) video summary generation, as depicted in Figure 1.

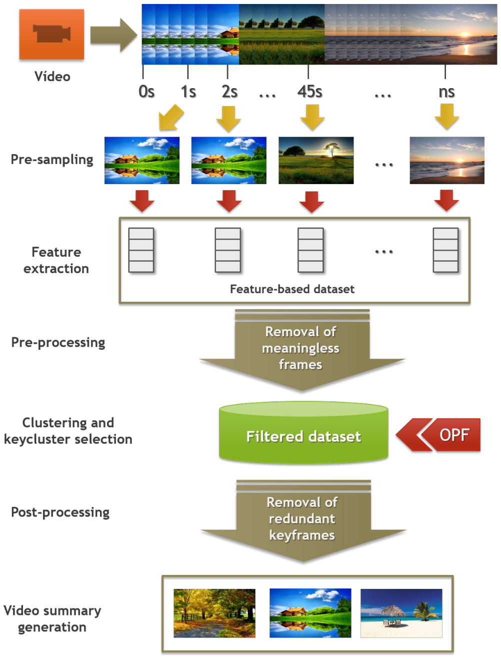


Fig. 1. Steps performed during video summarization.

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.

The second phase performs the *feature extraction* from each frame extracted in the previous step. In this work, we considered two descriptors to encode color information: Global Color Histogram (GCH) [12] and Color Coherent Vector (CCV) [13]. GCH was used in the Openvideo dataset, although CCV was responsible to encode color information from Youtube dataset. Therefore, after the feature extraction step, we have two feature-based datasets ready to be processed. Note we used two color descriptors, since Martins et al. [8] observed GCH and CCV work differently for each dataset.

Further, we performed the removal of meaningless frames from the feature-based dataset aiming at avoiding unnecessary frames during the clustering process. Note that a meaningless

[1] http://www.ffmpeg.org/

[2] http://sites.google.com/site/vsummsite/

[3] http://www.open-video.org/

[4] http://www.youtube.com/

frame is the one whose image is composed of a single color (i.e., full black or white frames) due to a fade-in or fade-out effects. Therefore, such frame is then removed from the feature-based dataset only if the color variance of its quantized image is equal to zero [3].

In the third step, OPF computes the clusters from the feature-based dataset aiming at finding the most representative frames on each cluster (keyframes). Since OPF finds the prototypes in the regions with highest density, they tend to be located at the center of the clusters, thus being good candidates to become keyframes. The main problem related to the approach proposed by Martins et al. [8] concerns the fact OPF was applied in the whole dataset, which means spatial-similar frames are associated to the very same cluster, but they may have no temporal relation to each other. Therefore, one may loose such kind of crucial information. In this paper, we propose to address this problem by partitioning the feature-based dataset into n smaller subsets to preserve the temporal and spatial information of each frame. Since OPF is now executed on each subset, one can also speed up the whole process by running several OPFs in parallel, given the learning process is executed independently. Figure 2 presents in more details the third step of our proposed method for video summarization.

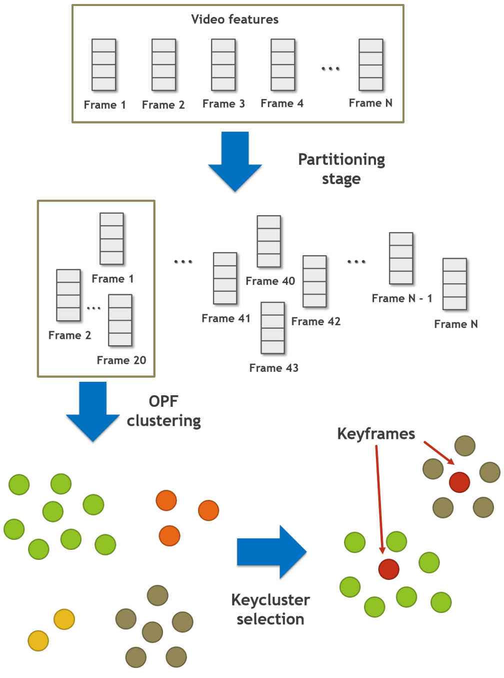


Fig. 2. Detailed explanation about OPF clustering process.

In order to grasp the synergy between temporal and spatial information, we proposed one more enhancement that is related to the OPF mechanism to compute the "distance" among frames. In this work, we designed a function that considers more information than just the spatial-content encoded by the Euclidean distance among the feature vectors of each frame, as presented by Martins et al. [8]. Such function is composed of two terms, being the first one related to the temporal information T_{ij} between frames i and j, and the another related to the spatial information S_{ij} among those frames. The temporal term is given as follows:

$$T_{ij} = \left| p_i - p_j \right|, \tag{3}$$

where p_i and p_j stand for the normalized position of frames i and j, respectively. Note the position of each frame denotes its chronological location in the video. Therefore, the normalized position is computed by just dividing the frame number with the total number of frames.

The spatial term is formulated as follows:

$$S_{ij} = \frac{d(i,j)}{d_{max}}, \tag{4}$$

where $d(i,j)$ stands for the Euclidean distance between frames i and j, and d_{max} denotes the maximum Euclidean distance among any two different frames. Finally, the proposed distance function is given by:

$$D_{ij} = S_{ij} + \alpha T_{ij}, \tag{5}$$

where α is a relaxation term that weights the amount of temporal information considered during the final distance computation[5].

Even after the clustering be performed on each subset, one can also have small clusters, which means they may not contribute with relevant information to the final video summary. In order to remove such non-relevant clusters, we compute the average cluster size for each subset, and then we keep the clusters whose size (number of samples that belong to it) is greater than the half of the average cluster size [4]. Soon after, we then extract one keyframe from each remaining cluster (*keycluster*), being such keyframe the prototype of that cluster. The collection of all keyframes composes the final frame set.

The fourth step is responsible for removing redundant keyframes from the frame set obtained in the previous phase. This process is described as follows: each keyframe is compared against all other keyframes using the Euclidian distance. If the resulting distance is smaller than 0.15, this keyframe is considered irrelevant, thus being removed from the summary. The threshold used for comparison purposes was selected empirically. In the final step, the keyframes are chronologically ordered to generate the video summary. Therefore, the final static summary can now be used for comparison purposes against others.

IV. Methodology and Experiments

In this section, we present the methodology and the experiments conducted to validate the proposed approach, hereinafter called OPF*. For comparison purposes, we evaluated OPF* against with the results reported by OPF [8], OV [14][6],

[5] Note this procedure is applied on each subset.

[6] Note the storyboards generated using the algorithm of DeMenthon et al. [14] and refined through some manual intervention.

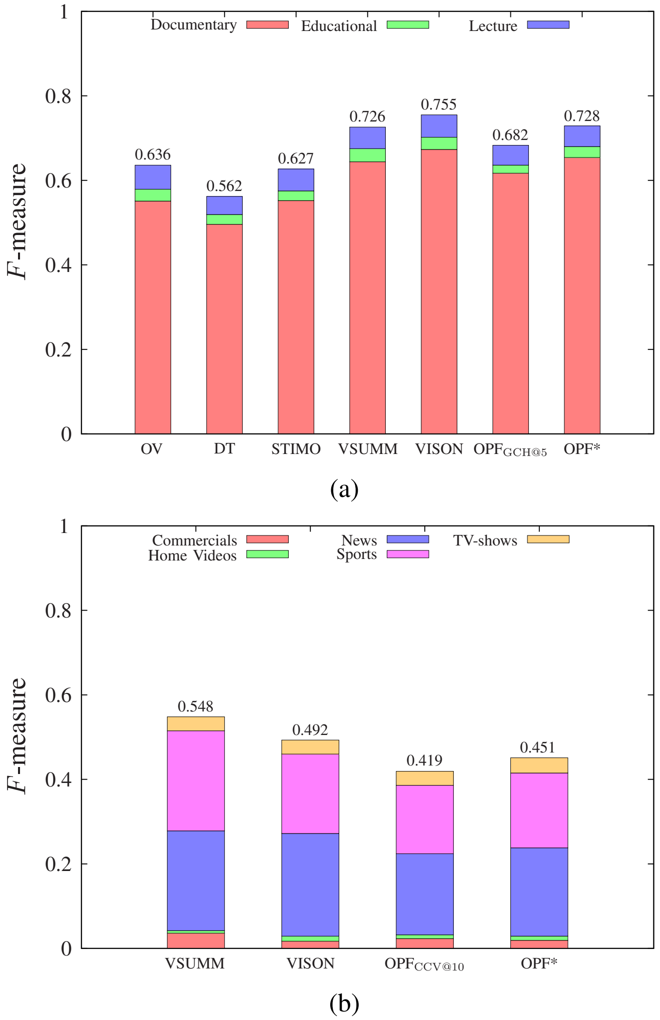
DT [15], STIMO [16], VSUMM [4], and VISON [3] concerning Open Video dataset. On the other hand, considering the Youtube dataset, OPF* was compared against OPF, VSUMM and VISON only, since the other approaches do not have results reported on this dataset.

In this work, we adopted a subjective evaluation method to determine the quality of video summaries, known as Comparison of User Summaries (CUS) [4], which works as follows: initially, the subjects are asked to watch the entire video, and further they are oriented to select a subset of frames which is able to summarize the video content. Note that each subject is free to select the number of frames to compose his/her summaries. Finally, their summaries are compared against the summaries generated by the algorithms. Besides, we used the pixel-wise matching method proposed by Almeida et al. [3] to compare frames from different summaries. 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. Finally, we employed the F-measure as the metric used for evaluating the performance, mostly due to the trade-off between precision and recall. It is noteworthy that F-measure is one of the most used approaches for video summaries analysis.

As aforementioned, OPF computes clusters on-the-fly based on the k_{max} variable (Section II), which defines the maximum number of nearest neighbors to be considered during cluster computation. Although the reader may argue OPF does have one parameter, 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. In order to select the best value of k for each subset, we conducted the following methodology: we tried different percentages of the subset sizes (15%, 20%, 25%, 30%, 35%, 40%, 50% and 60%), and for each one we evaluated $k_{max} \in [5, 50]$ with steps of 5. Finally, we selected the subset size and k_{max} that maximized the F-measure, i.e, 25% and $k_{max} = 5$ for both descriptors (GCH and CCV).

Figure 3 shows the F-measure values for all techniques and datasets considered in this paper. Clearly, OPF* obtained more accurate results than OPF for both datasets, as well as it has been the second best technique considering Open Video dataset (Figure 3a). Additionally, it has been placed as the third more accurate technique in the YouTube dataset (Figure 3b). However, the best technique in YouTube dataset uses k-means for clustering purposes, thus requiring the number of clusters beforehand. Note that information is not a main concern regarding OPF-based techniques. It is worth noting to stress that OPF requires less user interaction than VISON technique as well, since it has some user parameters.

We observed OPF* seems to work better with smaller subsets, since larger ones do not favor the temporal information. Additionally, $\alpha = 0.86$ (Equation 5) worked well for both datasets. In our experiments, we observed that small values for α did not contribute a lot for the final results. In regard to Open Video dataset, OPF* achieved better results than OPF concerning the "Documentary" and "Educational" videos. The rationale behind that concerns with the fact that "Educational" videos contain similar frames but at different temporal positions in the video. Imagine some lecturer teaching a specific subject, and further we may have some pictorial explanation about that, and once again the teacher gets focused again in the video. Although we have quite spatial-similar frames, they are placed at different temporal positions within the video.



(a)

(b)

Fig. 3. Mean F-Measure achieved by different approaches considering each video category for (a) Open Video and (b) YouTube datasets.

With respect to YouTube dataset, the best improvement regarding OPF* concerns with "Sports" videos, which are also expected to cover similar situations that have near-spatial frames, such as the best moments from a soccer game, for instance. Since we used color descriptors, it is very likely from this point of view that different soccer games seem similar to each other. Once again, the temporal information played an importante role in this situation.

V. Conclusions

In this work, we proposed a new approach for video summarization that allows OPF to consider both spatial and temporal information when clustering keyframes to compose the final summary. The proposed approach innovates in two parts: (*i*) a subset-driven clustering process, and (*ii*) a different distance function that considers both spatial- and temporal-like information from the video datasets.

We showed improvements with respect to the former OPF work on video summarization, as well as we obtained results very competitive to some state-of-the-art techniques for static video summarization in two public datasets. Our future works will consider specific methodologies for different video classes, aiming at increasing the global F-measure.

Acknowledgment

The authors would like to thank CNPq (grants #470571/2013-6 and #306166/2014-3) and FAPESP (grants #2014/16250-9 and #2015/50319-9) for their financial support.

References

[1] A. G. Money and H. W. Agius, "Video summarization: A conceptual framework and survey of the state of the art," *J. Visual Communication and Image Representation*, vol. 19, no. 2, pp. 121–143, 2008.

[2] J. Almeida, N. J. Leite, and R. S. Torres, "Online video summarization on compressed domain," *Journal of Visual Communication and Image Representation*, vol. 24, no. 6, pp. 729–738, 2013.

[3] ——, "VISON: VIdeo Summarization for ONline applications," *Pattern Recognition Letters*, vol. 33, no. 4, pp. 397–409, 2012.

[4] S. E. F. Avila, A. P. B. Lopes, A. Luz Jr., and A. A. Araújo, "VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method," *Pattern Recognition Letters*, vol. 32, no. 1, pp. 56–68, 2011.

[5] Y. S. Choi and K. J. Kim, in *International Conference on Computational Science and Its Applications*, ser. Lecture Notes in Computer Science, A. Laganá, M. L. Gavrilova, V. Kumar, Y. S. Mun, C. J. K. Tan, and O. Gervasi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, vol. 3043, ch. Video Summarization Using Fuzzy One-Class Support Vector Machine, pp. 49–56.

[6] D. P. Papadopoulos, A. A. Chatzichristofis, and N. Papamarkos, "5th international conference on computer vision/computer graphics collaboration techniques," ser. Lecture Notes in Computer Science, A. Gagalowicz and W. Philips, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, vol. 6930, ch. Video Summarization Using a Self-Growing and Self-Organized Neural Gas Network, pp. 216–226.

[7] L. M. Rocha, F. A. M. Cappabianco, and A. X. Falcão, "Data clustering as an optimum-path forest problem with applications in image analysis," *International Journal of Imaging Systems and Technology*, vol. 19, no. 2, pp. 50–68, 2009.

[8] G. B. Martins, L. C. S., D. Osaku, J. G. Almeida, and J. P. Papa, "Progress in pattern recognition, image analysis, computer vision, and applications," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, ser. Lecture Notes in Computer Science, E. Bayro-Corrochano and E. Hancock, Eds. Springer International Publishing, 2014, vol. 8827, ch. Static Video Summarization through Optimum-Path Forest Clustering, pp. 893–900, 19th Iberoamerican Congress on Pattern Recognition.

[9] C. Castelo-Fernández and G. Calderón-Ruiz, "Progress in pattern recognition, image analysis, computer vision, and applications," in *20th Iberoamerican Congress on Pattern Recognition*, ser. Lecture Notes in Computer Science, A. Pardo and J. Kittler, Eds. Springer International Publishing, 2015, vol. 9423, ch. Automatic Video Summarization Using the Optimum-Path Forest Unsupervised Classifier, pp. 760–767.

[10] D. Comaniciu, "An algorithm for data-driven bandwidth selection," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 281–288, 2003.

[11] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, Aug 2000.

[12] M. J. Swain and B. H. Ballard, "Color indexing," *Int. J. Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.

[13] G. Pass, R. Zabih, and J. Miller, "Comparing images using color coherence vectors," in *ACM Int. Conf. Multimedia (ACM-MM'96)*, 1996, pp. 65–73.

[14] D. DeMenthon, V. Kobla, and D. S. Doermann, "Video summarization by curve simplification," in *ACM Int. Conf. Multimedia (MM'08)*, 1998, pp. 211–218.

[15] P. Mundur, Y. Rao, and Y. Yesha, "Keyframe-based video summarization using Delaunay clustering," *Int. J. on Digital Libraries*, vol. 6, no. 2, pp. 219–232, 2006.

[16] M. Furini, F. Geraci, M. Montangero, and M. Pellegrini, "STIMO: STIll and MOving video storyboard for the web scenario," *Multimedia Tools Appl.*, vol. 46, no. 1, pp. 47–69, 2010.