Partial Least Squares Image Clustering

Ricardo Barbosa Kloss*, Marcos Vinicius Mussel Cirne†, Samira Silva*, Helio Pedrini†, William Robson Schwartz*
*Computer Science Department, Federal University of Minas Gerais, Belo Horizonte-MG, Brazil
†Institute of Computing, University of Campinas, Campinas-SP, Brazil
rbk@dcc.ufmg.br, marcosvcirne@gmail.com, samirasilva@dcc.ufmg.br, helio@ic.unicamp.br, william@dcc.ufmg.br

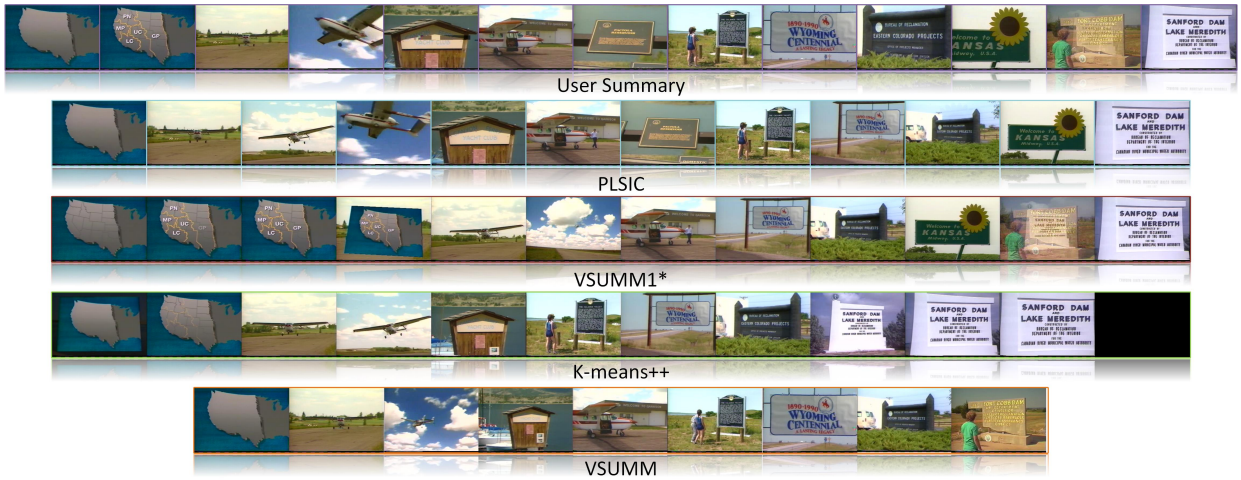

Fig. 1. A user summary, shown in the first row, represents a possible ground truth. The second row presents a summary obtained with the proposed approach, referred to as PLSIC. Third and fourth rows display summaries obtained by means of our implementation of VSUMM [1] technique and K-means++, where the number of clusters K is estimated by a shot boundary detection algorithm. The last row is a summary obtained by the original VSUMM method [1], publicly available.

***Abstract*—Clustering techniques have been widely used in areas that handle massive amounts of data, such as statistics, information retrieval, data mining and image analysis. This work presents a novel image clustering method called Partial Least Square Image Clustering (PLSIC), which employs a one-against-all Partial Least Squares classifier to find image clusters with low redundancy (each cluster represents different visual concept) and high purity (two visual concepts should not be in the same cluster). The main goal of the proposed approach is to find groups of images in an arbitrary set of unlabeled images to convey well defined visual concepts. As a case study, we evaluate the PLSIC to the video summarization problem by means of experiments with 50 videos from various genres of the Open Video Project, comparing summaries generated by the PLSIC with other video summarization approaches found in the literature. A experimental evaluation demonstrates that the proposed method can produce very satisfactory results.**

***Keywords*-Image Clustering; Partial Least Squares; Video Summarization; Shot Sampling.**

I. Introduction

An increasing volume of digital images and videos has become available over the years due to the growth of smartphones, tablets and Internet of Things in general. Therefore, the development of techniques capable of managing large amount of data in a fast and accurate way is important to extract any valuable information.

Finding natural groupings is the goal of clustering methods, such as K-means [2]. They can help classify and separate information in order to make data analysis easier. Examples of problems related to data grouping are data indexing, data compression and natural image classification [3]. Applied to visual concepts, Singh et al. [4] proposed a method for grouping image patches to separate visual concepts and then use them as mid-level features [5].

Video summarization is the task of selecting a set of images that compose a video, called frames, such that the selection can convey the main information of the video. This way, the user does not need to watch the entire video, but only those segments of particular interest, which helps dealing with the huge amount of data generated nowadays.

A classic method for extracting a summary from a video is by grouping a subset of its frames by means of a clustering algorithm, such as K-means [2]. However, the problem with this approach is that K-means does not work properly on general scenarios. For instance, only groupings that are circularly shaped will be found when Euclidean distance is used. Furthermore, the inherent problems related to the analysis of data in high dimensionality, called curse of dimensionality [6], make it very hard to separate the frames based on distance metrics [7].

This work proposes a novel image clustering method based on Partial Least Squares [8], called *Partial Least Squares Image Clustering* (PLSIC). Our method is inspired by the work developed by Singh et al. [4], which also focuses on improving the K-means algorithm for grouping images. Their method employs K-means to find an initial grouping of image patches, prunes unsatisfactory results and then, until convergence, it learns a binary SVM classifier for each cluster, which is then used to search for suitable patches that will be assigned to clusters. Since Singh et al. [4] employ SVM classifiers, a minimum number of samples and a set of negative samples are required (e.g., large set of images acquired from Flickr[1] for learning). On the other hand, we use the one-against-all Partial Least Squares (OAA-PLS) [9], which is capable of dealing with a limited number of samples and is suitable for video summarization, where negative samples are not available.

The main disadvantage of employing K-means to group images is that it can easily get stuck to local optima [3], which means that its clusters can be far from the optimal solution, *i.e.,* it can have low purity (mixed samples in a given cluster). Due to the ability of the OAA-PLS in capturing different visual information with highly unbalanced class distributions based on a single or very few samples per class [9], our approach can be initialized by using only the sample closest to the centroid of each cluster, which is expected to keep high cluster purity.

Regarding the video summarization problem, if the initialization of the clustering algorithm does not include a given keyframe, there is no guarantee that it will be added to the summary in later steps. This way, our approach can be initialized with a large number of pure clusters (frames) that might be merged in a later step to minimize the redundancy among clusters. As a consequence, the final clusters will be composed of discriminative visual concepts. Therefore, the goal of the proposed work is to find groups of images in an arbitrary set of unlabeled images to convey well defined visual concepts. In the context of video summarization, the goal is to find a grouping of frames that also captures a visual concept, often from a shot of the video, and then, from this grouping, pick one as a *keyframe*. This motivates us to apply this technique to video summarization.

Our method is evaluated on videos from the Open Video Project[2], a widely used dataset for video summarization, and compared to other techniques available in the literature, including the simple K-means approach, VSUMM summaries, and our implementation of VSUMM_1, which has no preprocessing step and uses the same pipeline as our method to have a fair comparison. The proposed PLSIC scored a F-measure [10] of 0.649 in the summarization of videos from the Open Video Project, which is a significant result since we do not employ any preprocessing to generate the summaries, a common strategy found in video summarization approaches.

[1]http://press.liacs.nl/mirflickr

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

II. Basic Concepts and Related Work

In this section, some of the main approaches related to clustering and video summarization are briefly presented and discussed, along with some basic concepts.

A. Data Clustering

A data clustering problem is one in which a set of elements must be grouped to minimize the dissimilarity of elements present in the same group and maximize the dissimilarity in elements that belong to different groups. A more in-depth discussion about all these problems can be found in Jain [3].

A classic clustering algorithm is K-means [2]. The standard K-means approach depends on three user-specified parameters: number of clusters K, cluster initialization and distance metric. The Euclidean distance is the commonly used similarity metric, which finds ball-shaped groupings of data. The initialization parameter can lead to very different results, since the method can converge to local optima. The estimation of the parameter K can be very difficult since it needs a previous detailed knowledge of the domain of the data to be clustered, which is proven to be even more difficult when considering high dimensionalities of data, typically found in many computer vision applications.

Besides K-means, there are various other clustering algorithms, which define similarity or connectedness between clusters differently. Some clustering methods, like DBSCAN [11], consider similarity as the number of common neighbors shared. Other clustering methods employ probabilistic mixture models, where each cluster is described by one or more mixture components, e.g., Expectation Maximization [12] and Latent Dirichlet Allocation [13].

B. Classification-based Clustering

Since clustering approaches applied to the image domain do not provide satisfactory results, due to high dimensionality and distance metrics, Singh et al. [4] proposed a clustering method that uses a classifier to refine and improve the separation obtained by K-means. The goal was to find patches of images that were discriminative of arbitrary visual concepts that could then be used as mid-level features. Our approach is inspired on theirs.

The method in [4] employs an iterative procedure that alternates between clustering and training of discriminative classifiers, while applying careful cross-validation at each step to prevent overfitting. The classifiers are used to increase the *purity* of the clusters, which is defined by homogeneity of visual concepts. Because they employ SVM classifiers, they require a minimum number of samples and a set of negative samples, for which they consider a large dataset composed by images acquired from Flickr. On contrast, we employ an OAA-PLS, which does not require the need for both a minimum number of samples and a negative set.

A disadvantage of the approach proposed by Singh et al. [4] is its need for large datasets of images, a set of interest, known as the discovery set, and another to represent the "natural world". Singh et al. [4] used this "natural world" as

negative set to learn the SVM classifiers, i.e., for each cluster, a classifier was learned to discriminate the visual concept of the cluster from the natural world. In addition, their use of SVM implies that the clusters need to have a minimum size since SVM does not work well with a small number of samples.

The demand for a large negative set in [4] makes their method unsuitable for video summarization. This happens due to the fact that the data consist of frames from a video and the goal is to classify them in an unsupervised way, where the definition of a negative set would require a previously acquired information. On the other hand, due to the use of an OAA-PLS classifier, our method does not need a negative set since the classification focuses on discriminative features by employing a one-against-all classification scheme.

C. Video Summarization

A video consists of images, named frames, which can be semantically grouped into shots. Thus, a shot is a collection of similar frames, that share a common visual concept. Physically, a shot is often the product of a contiguous recording of a video, and therefore, its frames are usually from one same location and conveying one same motif. Furthermore, the shots can be semantically grouped into scenes, composing the hierarchy of a video.

The task of video summarization can be defined by finding a set of keyframes with low redundancy and discriminative important events in the video. Video summarization techniques can be divided into static and dynamic. In the former, the summary is a series of still images (keyframes) and are a representation of a part of the video content, whereas, in the latter, frames from the video are selected to compose short clips or a video skim. Truong et al. [14] consider to be more entertaining watching a skim than a slide show of keyframes. They also assert that keyframes can be, for instance, reordered to show spatial relationships instead of being chronologically ordered, which can be more representative and also reduce computational cost for various video analysis and retrieval applications.

Clustering algorithms are commonly employed to separate similar frames into groups and choose a frame from each of these groups as a keyframe, as can be seen in Mundur et al.; Furini et al. and Avila et al. [15], [16], [1]. A popular method is the spectral clustering, which also can be used in video summarization to extract keyframes [17] and for shot boundary detection [18].

Mundur et al. [15] proposed a video summarization method based on Delaunay triangulation [19] to make an automatic cluster of video keyframes. For each video frame, an HSV-color histogram of 256 bins is constructed. From this histogram, a 256-dimension line-vector is created. The composition of all line-vectors from all video frames forms then a matrix of dimensions $N \times 256$, where N is the total number of frames. Next, a Principal Component Analysis (PCA) [20] is applied to reduce the dimensionality of this matrix, optimizing the total processing time. Then, the Delaunay triangulation algorithm is executed on the data to generate the appropriated clusters. The keyframes obtained from each cluster are then identified from their respective centroids.

Furini et al. [16] developed a video browsing system with a summarization technique that produces both static and dynamic summaries on-the-fly, where users can customize summary properties such as storyboard length and the maximum time the system must take to produce the storyboard. This technique uses a fast clustering method that takes frame characteristics from HSV color space to separate the frames of a video in different groups. In doing so, this approach becomes faster than standard K-means, as well as the method of [15].

Avila et al. [1] proposed a methodology to produce static video summaries. Instead of using all frames of a video, the method samples the video to only one frame per second, reducing the summarization processing time. Furthermore, the feature extraction process is done by using 16-bin color histograms of the hue component of the HSV-color space frames. Then, the number of clusters is estimated by computing the pairwise distances between consecutive frames and comparing then to a fixed threshold. Once this number is found, the frames are clustered and the summary is generated. Moreover, they also developed an evaluation metric called CUS (Comparison of User Summaries), in which, for each video used in the tests, the automatic summaries (the ones generated by their method) are compared to five manual summaries (ground-truth), produced by different users. Later, similar frames between the automatic summary and the manual summaries are identified, based on the same HSV-color space histogram used in the feature extraction process.

Almeida et al. [21] presented an online summarization approach named VISON, which handles video frames on the compressed domain and allows user interaction and customization of specific parameters, such as quality of summaries and maximum waiting time. In this approach, video frames are reduced to DC images (based on Discrete Cosine Transform (DCT) coefficients) and HSV-color histograms are then extracted from these images.

Mahmoud et al. [22] combined color and texture features from video frames to generate summaries. In addition, the frames are clustered by a modified version of the DBSCAN clustering algorithm [23], selecting the middle core frame of each cluster as the keyframes that comprise the final summary. As in [1], the CUS metric was used to evaluate the quality of the summaries, but the similar frames were detected using both color and texture features, instead of using only color.

The proposed image clustering approach is similar to the method presented by Singh et al. [4], although there is no need for a negative set, which makes our method more general and also possible to be applied in video summarization. The merge step in our approach can also reduce redundancy. In addition, since our method compares entire clusters, it is less costly than the pairwise frame comparison as employed in [1], [21].

The flow of of execution of our experiments are similar to that developed by Avila et al. [1], in which there is an initialization step followed by a grouping of images and, then, a step to reduce redundancy of the summary. Although

Avila et al. [1] rely on K-means [2], it is not well suited for high dimensionality grouping, as it is the case of image data. Our approach, though dependent on the cluster initialization, reduces its dependency from K-means by performing a later merge step.

III. Proposed Approach

The main purpose of our work is to find groups of images in an arbitrary set of unlabeled images (the discovery dataset D) in such a way that the elements in these groups can be discriminative and, therefore, convey a well defined visual concept. The flow of execution of our proposed method is illustrated in Figure 2 and summarized as follows.

Initially, the method receives an input of extracted features (the discovery set) and splits it in two subsets (D_1 and D_2) to avoid overfitting of the classifiers. Then, an initial grouping is estimated on D_1 by using some clustering method (Section III-A). Afterwards, an OAA-PLS classifier is trained based on the initial clustering and this classifier is used to search for elements of high response in the D_2 subset (Section III-B). These elements are assigned to new clusters (Section III-C), which are compared against each other for similarity, such that similar clusters are merged (Section III-D). After the merge step, a new classifier is trained on the new clusters composed of D_2, whereas elements that belong to D_1 are assigned to form new clusters. These steps are performed until convergence, which can occur due to a maximum number of iterations, a minimum number of clusters, lack of merging or cluster stability.

A. Initialization

The initialization is composed of two steps: construction of the discovery subsets and assignment of the initial clusters. First, it splits the input dataset (discovery set) in two equal sized disjoint sets, called D_1 and D_2. Then, a clustering method may be applied on D_1 in order to obtain an initial clustering. This initial clustering does not need to be very pure. However, its purity will affect other parameters of the method. Some methods for initialization are evaluated and discussed during the experiments.

Similarly to Singh et al. [4], the discovery set contains the samples to be clustered. However, since an OAA classifier is employed, there is no need for a natural set containing counter-samples, which is an advantage of our method and what makes it applicable to video summarization.

B. Learning

An OAA-PLS classifier is learned for each cluster available. Since it is a one-against-all classifier, for each class/cluster there is a training stage in which the elements in that class are considered to be from a positive class and the elements of other classes are considered to be negative. This way, each classifier will focus on features that better discriminate the positive class from the remaining classes, in other words, focusing on increasing the purity of each cluster.

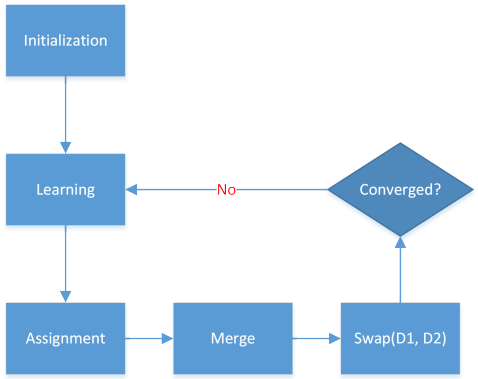

Fig. 2. Flowchart of the proposed clustering method.

By using a one-against-all PLS classifier, we are able to capture different visual information with highly unbalanced class distributions with a single or very few samples in the positive class [9]. Therefore, our method eliminates the need for a negative set, which is suitable for video summarization, and also allows the use of a single element in each of the initial clusters, which assures high cluster purity.

C. Assignment

Since an OAA classifier is employed, if an element belongs to two different classes, the learning of the classifier can be affected. For this reason, a hard assignment technique was chosen, that is, an element can only be assigned to a single cluster.

In the assignment step, a matrix is constructed in which an element $m_{i,j}$ represents the response of the j-th dataset entry against the i-th class of the classifier. Each row of this matrix is sorted according to the responses, but without losing information of the original position which represents the identification of the sample. Then, for each row, the value of the first m elements that are not yet assigned are added. The row with the largest sum is chosen and its first m unassigned elements are assigned to a new cluster. New clusters are formed until the number of new clusters is the same as the number of old clusters.

D. Merge

The merge was inspired by *agglomerative hierarchical clustering* [3]. It consists of two steps: construction of a similarity matrix and search for similar clusters. The similarity matrix is a structure in which every element $M(i, j)$ is a similarity value, according to some similarity metric between the i-th and j-th clusters.

After constructing the similarity matrix, similar clusters are searched. Since the position in the matrix with the highest value represents the most similar pair of clusters, it is selected. If it is higher than a merge threshold, λ, the clusters corresponding to the row and column of the selected position are merged, that is, one of them is removed and the other receives the elements of the deleted cluster and the row and column that participated in the merge are assigned as zero value, so they

do not interfere in further calculations. Then, another position in the matrix is selected. This process is repeated until there is no position with a value higher than λ (note that no cluster can participate in more than one merge; therefore, at a given iteration of the method, the number of clusters can be reduced to half its size at most).

The merging step, as described, is able to eliminate redundant clusters, improving the results achieved by K-means, as it will be shown in the next section, where we compare the results achieved by the proposed approach with video summaries obtained using only K-means.

IV. Experiments on Video Summarization

Even though the proposed image clustering approach can be used in more general purpose problems, we demonstrate its application to the video summarization problem since, as mentioned earlier, our method is able to find groups of images in an arbitrary set of unlabeled images conveying well defined visual concepts. Elements from the same shot tend to have one same visual concept, therefore, by finding discriminative visual concepts, the method can also find different shots, hence, being suitable for video summarization. A shot corresponds to abrupt video frame transitions related to the searched visual concepts, making our method suitable for video summarization.

Figure 3 illustrates the steps to perform video summarization and how they relate to the proposed clustering algorithm, the PLSIC. The evaluation, feature extraction and the post-processing are discussed in Sections IV-A, IV-B and IV-C, respectively. The experimental setup is described in Section IV-D and the results obtained using the Open Video Project dataset are presented and discussed in Section IV-E.

A. Evaluation of Video Summarization

Evaluating video summaries is a very difficult task, especially when there is no objective ground-truth to compare the results against. Some attempts have been proposed in the literature to develop a consistent framework for objective evaluation of video summaries, as can be seen in Avila et al. [1] and Almeida et al. [21].

The evaluation proposed by Avila et al. [1] is performed as follows. Subjects are asked to make summaries of all videos in the dataset. To do that, they are asked to select a subset of frames that is able to summarize the video content. Each subject can select any number of frames. Finally, their summaries can be compared against summaries produced automatically.

The algorithm for finding matchings relies on a pixel-wise comparison. Corresponding pixels are considered as different if their intensity values differ at least in one of their corresponding 4-neighbors. The descriptor used in the comparisons is the Color Co-occurrence Matrices (CCM) [24], [25], [26]. The similarity of two frames is then the ratio of number of similar pixels to the total number of pixels. Two frames are matched if their similarity on the Normalized Sum of Square Distance (NSSD) metric [27] is greater than a threshold value, which was defined as 0.2. In the employed evaluation

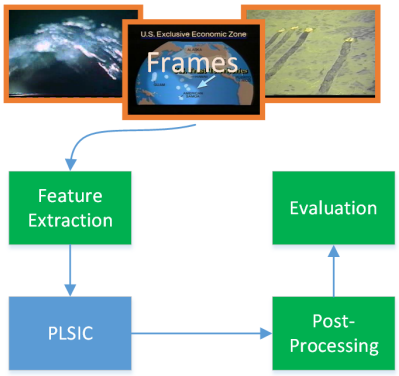

Fig. 3. Flowchart of the experiments.

method, frames from the automatic summary are compared against frames from the user's summary. When a match occurs, the matching frames are removed from the next iterations of comparisons.

Precision and Recall are the usual quality metrics in video summarization. Precision is the ratio of matched frames in the automatic summary divided by the number of frames in the automatic summary, whereas Recall is the ratio of matched frames in the automatic summary divided by the number of user frames. These two measures have a trade-off relationship, such that an increase in precision usually decreases the recall. With that in mind, the evaluation metric chosen, the F-measure, combines precision and recall by means of harmonic mean into a single measure [10]

$$F = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

This is the same metric employed by Almeida et al.[21], rather than using the Comparison of User Summary (CUS) metric, proposed by Avila et al. [1].

B. Feature Extraction

To extract features from the frames, we employed the Color Co-occurrence Matrices (CCM), a derivation of the Gray Level Co-occurrence Matrices (GLCM) [28], [29], which have been widely used to extract many texture features, such as contrast, correlation, energy, entropy and homogeneity. Analogously, Color Co-occurrence Matrices [24], [25], [26] can be used to represent the distribution of color features between pairs of pixels in an image, considering the correlations between the color bands as well.

The construction of the CCM's from a color image I proceeds as follows: let $C_1, C_2, ..., C_n$ be the n channels of I, where each one is coded on L levels, and l the number of rows and columns of the CCM's, where l must be a divisor of L. Also, let C_u and C_v be a pair of channels (with $1 \leq u, v \leq n$). Finally, let $p = (x, y)$ be a pixel in I and $q = (x+\Delta x, y+\Delta y)$ a translation of p, such that q remains in the spatial domain of I. The computation of each position (i, j) of the CCM of size $l \times l$ and a translation vector $t = (\Delta x, \Delta y)$ for a pair of

channels C_u and C_v is done according to

$$\text{CCM}^t_{(C_u,C_v)}(i,j) = \text{card}\left\{\{p,q\} \in \mathbb{R}^2 \;\middle|\; \frac{C_u(p)}{l} = i, \frac{C_v(q)}{l} = j\right\},$$

where i and j range from 1 to l.

The video frames were represented through the RGB color space. Furthermore, $t = (1, 0)$ (one pixel to the right) and $l = 8$ (corresponding to a CCM size of 8×8). Since $\text{CCM}^t_{(C_u,C_v)}$ and $\text{CCM}^t_{(C_v,C_u)}$ store the same information, there are only 6 possible pairs of channels (C_u, C_v). Therefore, 6 different CCM's are constructed: (R,R), (R,G), (R,B), (G,G), (G,B) and (B,B). The final feature vector has 384 dimensions.

C. Post-Processing

This step is commonly employed in video summarization methods in order to refine the method precision, since some redundant keyframes may still be present.

After selecting the keyframes, a redundancy elimination algorithm is executed to discard similar keyframes. In this procedure, the similarity between all pairs of keyframes are analyzed. If the similarity of a given pair of keyframes is above a given similarity threshold T_S, one of these keyframes is discarded. The remaining frames will then make part of the final summary. Here, the NSSD [27] was used as the similarity function, which has been proved to be very robust and widely used in tasks that deal with digital image correlation [30]. This function ranges from 0 to 1, where the closer to zero, the more similar are the images. In our experiments, T_S was set to 0.2.

D. Dataset and Experimental Setup

The proposed approach has three main parameters: *initialization method*, *similarity metrics*, and *merge threshold* which are detailed as follows.

Initialization methods. Experiments are performed with three initialization methods: *random selection*, *K-means* and *shot sampling*. First, the initial clusters produced by random selection (RS) are composed by only one element, each randomly chosen, resulting in disjoint clusters. Second, the K-means initialization (KmI) is inspired by the approach proposed by Avila et al. [1], in which they explore the fact that frames have a temporal ordering. We execute K-means with a initialization set in which every five frames are assigned to a grouping, and then the K-means method is executed normally. Third, the shot sampling (SS) initialization consists of using an estimation of shot transitions and choosing random samples around the middle of the estimated shots.

Similarity metrics. To compare clusters for similarity, two similarity metrics are tested: the *cosine distance* of the cluster centroids and one that was inspired by *One Shot Similarity* [31]. These metrics are defined as

$$\text{oss\_like}_{i,j} = \frac{\sum_k^m R_{j,k} + \sum_k^m R_{i,k}}{2m}, \quad (1)$$

$$\text{cosine\_similarity}_{i,j} = \frac{Cn_i \bullet Cn_j}{\|Cn_i\|\|Cn_j\|}, \quad (2)$$

where R is a response matrix in which element $R(i, j)$ is the response of the j-th element against the i-th classifier, whereas Cn_i is the centroid of the i-th cluster.

Merge threshold. The merge threshold, λ, is a value that defines how similar two clusters need to be in order to be considered redundant. A small grid search was performed to find an optimal merge threshold, in which the values $[0.3, 0.5, 0.7]$ were experimentally set.

E. Results on the Open Video Project

The method is evaluated on videos from the Open Video Project. There are 50 videos, all in MPEG-1 format (with size of 352×240 pixels and 29.97 frames per second), in color and with sound. They are distributed among several genres (e.g., documentary, educational and lecture) with duration varying from 1 to 4 minutes. The videos are the same used in Mundur et al. [15], Furini et al. [16], Avila et al. [1] and Almeida et al. [21].

TABLE I
MEAN F-MEASURE FOR EACH CONFIGURATION OF THE PLSIC METHOD.

Metric	λ	Initialization	mean F-measure	
			simple	post-processing
OSS	0.3	RS	0.567	0.610
OSS	0.5	RS	0.549	0.581
OSS	0.7	RS	0.496	0.592
Cosine	0.3	RS	0.619	0.637
Cosine	0.5	RS	0.621	0.635
Cosine	0.7	RS	0.632	**0.649**
OSS	0.3	SS	0.538	0.586
OSS	0.5	SS	0.549	0.597
OSS	0.7	SS	0.496	0.592
Cosine	0.3	SS	0.623	0.639
Cosine	0.5	SS	0.625	0.638
Cosine	0.7	SS	0.629	0.639
OSS	0.3	KmI	0.574	0.620
OSS	0.5	KmI	0.576	0.610
OSS	0.7	KmI	0.576	0.610
Cosine	0.3	KmI	0.622	0.638
Cosine	0.5	KmI	**0.633**	0.648
Cosine	0.7	KmI	0.631	0.644

Table I shows the results of different configurations of our method. Regarding the initialization methods, it is possible to see that all their results are very similar. However, the best result was obtained using Random Selection (RS) when combined with the post-processing step. We believe the reason why RS outperformed the other initialization methods is because it starts with only one element per cluster, and therefore, presenting high purity. Even though the mentioned initialization outperformed the other evaluated methods, we believe that it is highly dependent on the application, since the random initialization gives no guarantee of a good choice of the initial visual concepts.

According to Table I, the cosine similarity metric outperformed the OSS-like metric. The OAA Classifier ends up being unsuited for finding redundant clusters. In its learning step, if there are some clusters similar to others, then the

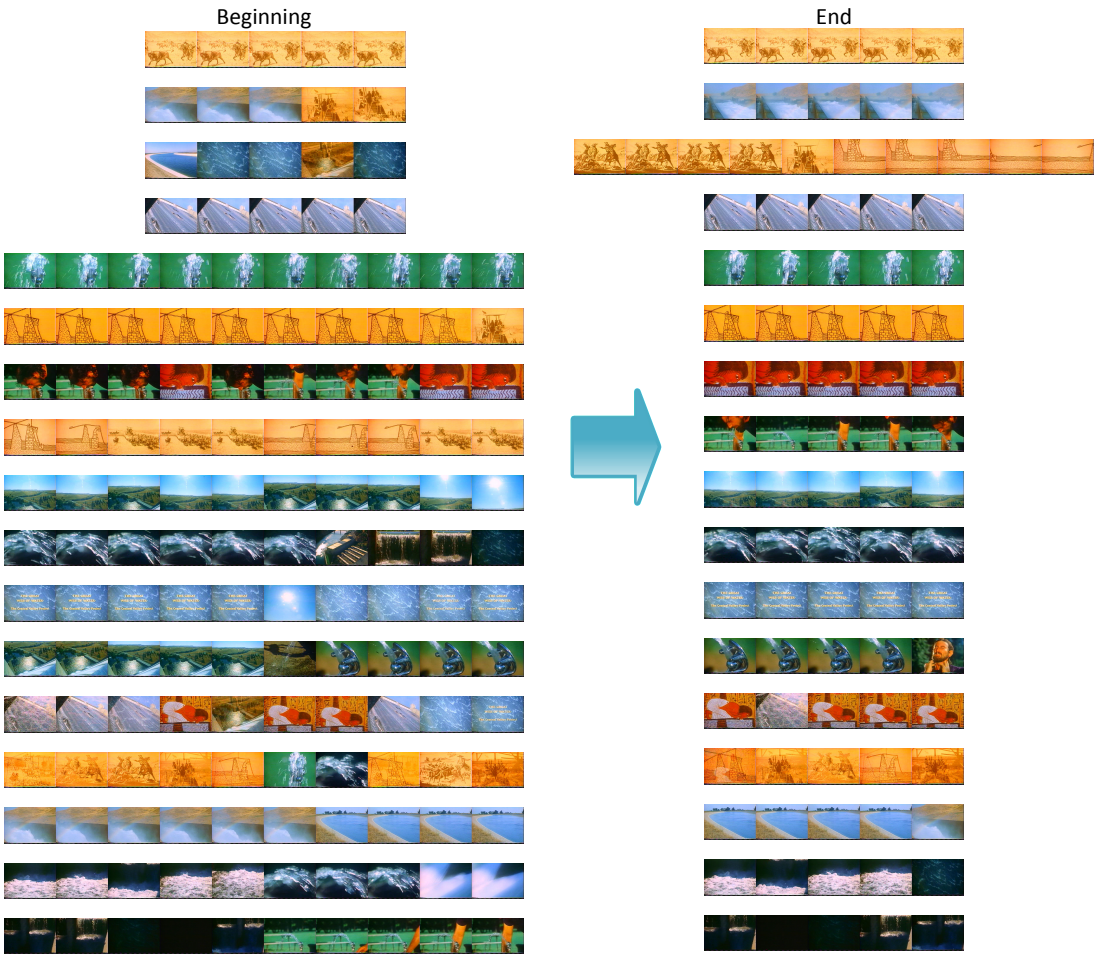

Fig. 4. Illustration of the progress of the method. The images on the left show clusters after the first iteration and images on the right show the clusters after the last iteration (each row shows one cluster - the clusters showed in the same row for the different iterations are not necessarily corresponding due to operations executed by the PLSIC). We can see a purity refinement in the clusters. The visual concept for each cluster is clearer after the last iteration.

classifier will be trained to differentiate them, since they have different labels. For this reason, when posed with two similar samples, whose visual concept belongs to one or more redundant clusters, the classifier will evaluate them as different, once it was trained to do so. Although the cosine metric performed better, there is a trade-off. Since the metric compares two clusters based on their centroids, it is not biased by any previous step of the method. On the other hand, the centroid is not suitable for representing a cluster with samples that are far distant from the mean, that is, a cluster with low purity.

A higher merge threshold implies that clusters will be considered redundant less often and, as a consequence, fewer merges will take place. In our experiments, a threshold of 0.7, often scored better before the post-processing step. Since it is a high threshold, some redundant clusters might not have been discarded by the method. On the other hand, the merge has less chance to mistakenly merge clusters of different visual concepts.

Figure 4 illustrates the progress of our method, showing some clusters obtained after setting the first assignment and merge, as well as the resulting clusters after the last assignment and merge. It is possible to observe that the purity of the clusters was significantly improved after 15 iterations. Even though there are still some clusters with impurities in the last iteration, they have very well defined visual concepts.

Table II shows the results achieved by different methods on the Open Videos Dataset. It is important to notice that the focus of our work is on the image grouping and, for that reason, we implemented a version of VSUMM [1], referred to as VSUMM$_1$*. This implementation allows a fair comparison of their clustering approach with ours without taking into consideration some implementation details specific for video summarization. The differences between VSUMM$_1$* and VSUMM are the lack of a preprocessing step, the features (CCM [24], [25], [26] instead of HSV color histogram) and the similarity metric (NSSD [27] instead of the Euclidean distance of the color histograms used in VSUMM). According to the results in Table II, the PLSIC was able to outperform VSUMM$_1$*.

As expected, our method outperformed the K-means [2] and the K-means++, since K-means is not suited for image

TABLE II
MEAN F-MEASURE ACHIEVED BY DIFFERENT APPROACHES.

K-means	K-means++	VSUMM	VSUMM$_1$*	PLSIC
0.478	0.530	0.730	0.623	0.649

grouping, once the distance metrics work poorly in high dimensionality. In addition, even though our method was not able to outperform the original VSUMM approach, it achieved results compatible to those obtained by VSUMM, which is a method specifically designed for video summarization, differently from ours that is a general image clustering approach without much tuning for this application.

V. CONCLUSIONS AND FUTURE WORK

In this work, we proposed and evaluated a new method for grouping images that can outperform K-means and replace it for image problems where there is redundancy in the dataset. Although its application in video summarization did not outperform the VSUMM [1] technique in our evaluation framework, our results are compatible to other approaches available in the literature.

In our implementation of the VSUMM technique, which it is submitted to the same scenario as our method, our clustering approach was able to outperform it. We conjecture that it is still possible to improve our results by tuning the parameters used in the method and in the OAA-PLS [9].

Some directions for future work include a further investigation on the initialization and similarity metrics employed in the method. Furthermore, we intend to apply our method to other problems that use clustering algorithms.

ACKNOWLEDGMENTS

The authors would like to thank the Brazilian National Research Council – CNPq (Grants #307113/2012-4, #487529/2013-8 and #477457/2013-4), Minas Gerais Research Foundation – FAPEMIG (Grant APQ-00567-14), São Paulo Research Foundation – FAPESP (Grant #2011/22749-8) and the Coordination for the Improvement of Higher Education Personnel – CAPES (DeepEyes Project).

REFERENCES

[1] S. E. F. de Avila, A. P. B. Lopes, A. da Luz Jr., and A. de Albuquerque 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.

[2] S. Lloyd, "Least squares quantization in pcm," *IEEE Trans. Inf. Theor.*, vol. 28, no. 2, pp. 129–137, Sep. 1982.

[3] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recogn. Lett.*, vol. 31, no. 8, pp. 651–666, Jun. 2010.

[4] S. Singh, A. Gupta, and A. A. Efros, "Unsupervised discovery of mid-level discriminative patches," in *European Conference on Computer Vision*, 2012.

[5] G. Csurka, C. Bray, C. Dance, and L. Fan, "Visual categorization with bags of keypoints," *Workshop on Statistical Learning in Computer Vision*, pp. 1–22, 2004.

[6] R. Bellman and R. Corporation, *Dynamic Programming*, ser. Rand Corporation research study. Princeton University Press, 1957.

[7] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is "nearest neighbor" meaningful?" in *In Int. Conf. on Database Theory*, 1999, pp. 217–235.

[8] G. M.-A. Morales, "Partial least squares (pls)methods: origins, evolution and application to social sciences," *Communications in Statistics - Theory and Methods*, vol. 40, no. 13, pp. 2305–2317, April 2011.

[9] W. R. Schwartz, H. Guo, J. Choi, and L. S. Davis, "Face Identification Using Large Feature Sets," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2245–2255, 2012.

[10] H. M. Blanken, A. P. d. Vries, H. E. Blok, and L. Feng, *Multimedia Retrieval (Data-Centric Systems and Applications)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.

[11] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, "Density-based clustering in spatial databases: The algorithm gdbscan and its applications," *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 169–194, Jun. 1998.

[12] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society: Series B*, vol. 39, pp. 1–38, 1977.

[13] D. Blei, A. Ng, and M. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, no. 4-5, pp. 993–1022, 2003.

[14] B. T. Truong and S. Venkatesh, "Video abstraction: A systematic review and classification," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 3, no. 1, Feb. 2007.

[15] P. Mundur, Y. Rao, and Y. Yesha, "Keyframe-based video summarization using delaunay clustering," *Int. J. Digit. Libr.*, vol. 6, no. 2, pp. 219–232, Apr. 2006.

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

[17] V. Chasanis, A. Likas, and N. Galatsanos, "Video rushes summarization using spectral clustering and sequence alignment," in *Proceedings of the 2nd ACM TRECVid Video Summarization Workshop*. New York, NY, USA: ACM, 2008, pp. 75–79.

[18] M. Guillemot, J.-M. Odobez, and D. Gatica-Perez, "Algorithms for video structuring," IDIAP, Idiap-Com Idiap-Com-05-2002, 0 2002.

[19] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*, 3rd ed. Springer-Verlag, 2008.

[20] S. Wold, K. Esbensen, and P. Geladi, "Principal Component Analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1–3, pp. 37–52, 1987.

[21] J. Almeida, N. J. Leite, and R. da S. Torres, "Vison: {VIdeo} summarization for {ONline} applications," *Pattern Recognition Letters*, vol. 33, no. 4, pp. 397 – 409, 2012.

[22] K. M. Mahmoud, M. A. Ismail, and N. M. Ghanem, "VSCAN: An Enhanced Video Summarization Using Density-Based Spatial Clustering," in *Lecture Notes in Computer Science*, vol. 8156. Springer, 2013, pp. 733–742.

[23] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *Second International Conference on Knowledge Discovery and Data Mining*. Portland, Oregon: AAAI Press, 1996, pp. 226–231.

[24] V. Arvis, C. Debain, M. Berducat, and A. Benassi, "Generalization of the Coocurrence Matrix for Colour Images: Application to Colour Texture Classification," *Image Analysis & Stereology*, vol. 23, no. 1, pp. 63–72, 2011.

[25] M. B. Islam, K. Kundu, and A. Ahmed, "Texture Feature Based Image Retrieval Algorithms," *International Journal of Engineering and Technical Research*, vol. 2, pp. 170–173, Apr. 2014.

[26] A. L. Lavanya and R. Sreepada, "A Generic Frame Work for Image Data Clustering Via Weighted Clustering Ensemble," *International Journal of Computer Science & Information Technologies*, vol. 3, pp. 5429–5433, Nov. 2012.

[27] B. Pan and Z. Wang, "Recent Progress in Digital Image Correlation," in *Application of Imaging Techniques to Mechanics of Materials and Structures, Volume 4*, ser. Conference Proceedings of the Society for Experimental Mechanics Series. Springer New York, 2013, pp. 317–326.

[28] R. Haralick, K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 3, no. 6, pp. 610–621, Nov. 1973.

[29] M. Unser, "Sum and Difference Histograms for Texture Classification," *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 8, no. 1, pp. 118–125, Jan. 1986.

[30] F. Hild and S. Roux, "Comparison of Local and Global Approaches to Digital Image Correlation," *Experimental Mechanics*, vol. 52, no. 9, pp. 1503–1519, 2012.

[31] L. Wolf, T. Hassner, and Y. Taigman, "The one-shot similarity kernel," in *IEEE International Conference on Computer Vision*, Sept. 2009.