Video Fingerprinting Using Latent Dirichlet Allocation and Facial Images

Nicholas Vretos, Nikos Nikolaidis and Ioannis Pitas
Department of Informatics, Aristotle University of Thessaloniki
Thessaloniki 54124, Greece Tel,Fax: +30-2310996304
e-mail: {nikolaid, pitas}@aiia.csd.auth.gr

Abstract

This paper investigates the possibility of extracting latent aspects of a video in order to develop a video fingerprinting framework. Semantic visual information about humans, more specifically face occurrences in video frames, along with a generative probabilistic model, namely the Latent Dirichlet Allocation (LDA), are utilized for this purpose. The latent variables, namely the video topics are modeled as a mixture of distributions of faces in each video. The method involves also Scale Invariant Features Transform (SIFT) based clustering of detected faces and adapts the bag-of-words concept into a bag-of-faces one, in order to ensure exchangeability between topics distributions. Experimental results provide evidence that the proposed method performs very efficiently for video fingerprinting.

1. Introduction

Video fingerprinting [1], also known as content-based copy detection, or robust perceptual hashing [2], or near replica detection [3], refers to methods that try to identify whether a given video is a replica or a near replica of one of the videos existing in a video database. It can be used in applications like digital rights management and copyright protection, multimedia databases management, broadcast monitoring etc. The need for efficient video fingerprinting algorithms is due to the enormous amount of video content and the scale of illegal video copying and distribution. Video sharing web sites such as YouTube need such algorithms in order to automatically check the intellectual property rights for videos that are uploaded in their database.

Many methods exist for image perceptual hashing [4]-[8]. However, their extension to video data (e.g. on a frame-by-frame basis) is not straightforward and efficient due to temporal dimension problems. A limited number of video fingerprinting or replica detection techniques have been proposed in the literature so far. In [9], Indyk et al. have used temporal fingerprints based on the shot boundaries of a video sequence in order to find pirated videos on the Internet. Oostveen et al. have proposed a spatio-temporal fingerprint based on luminance diference in spatiotemporal blocks [10]. B. Coskun et al. have proposed two robust video hashing algorithms for copy identification that are based on the Discrete Cosine

Transform (DCT) [11]. Hampapur and Bolle, have compared various global video descriptors based on motion, color and spatio-temporal intensity distribution [12]. Law-To et al. have proposed a technique for video copy tracking which is based on labels of local descriptor behavior computed along the video [13]. Their aim was to distinguish copies between highly similar videos, as well as to link similar videos, in order to reduce redundancy in video collections or to gather the associated metadata. Changick and Vasudev have proposed a copy detection scheme, where each video frame is partitioned into 2×2 blocks by intensity averaging [14]. Their spatiotemporal approach combines spatial matching of ordinal signatures obtained from the partitions of each video frame and temporal matching of signatures from the temporal partitions trails. Finally Lee and Yoo have presented a video fingerprinting method based on 2-D Oriented Principal Component Analysis (2D-OPCA) of affine covariant regions [15]. According to this method, in order to achieve robustness against geometric transformations, the fingerprints are extracted from local regions, covariant with a class of affine transformations. For reliable local fingerprints matching, only spatio-temporally consistent matches are taken into account.

Latent Dirichlet Allocation (LDA) is a generative probabilistic model introduced in [16]. It is a powerful method for capturing statistical properties of a collection of conditionally independent and identically distributed random variables.

The main idea behind LDA is the fact that such a set of random variables can be represented by a mixture of probability distributions. The latter is known as the de Finetti theorem [17]. This approach has already been applied in text modeling with good results [18]. It has been proven that LDA performs better than the pLSI (probabilistic Latent Semantic Indexing) algorithm, in the context of text modeling [16]. Moreover, this framework has been recently used in the context of image [19]-[25] and video analysis and description [26]-[30].

The novelty of this paper lies mainly in the use of latent aspects of the video content, aiming at extracting the underlying video topics and using them in video fingerprinting. In more detail, this paper includes the following novelties:

- The use of face occurrences in a video, to be called "facewords", that describe this video. However, this framework can be easily extended to cases without humans, since "facewords" can replaced by animals, even objects and scene artifact, provided that an adequate detector is used.

- The use of latent semantic analysis for video fingerprinting. Although many papers are using probabilistic Latent Semantic Analysis (pLSA) for a number of image and video processing, only very recent publications like [19] and [28] have utilized the LDA algorithm. However, none did use this framework for video fingerprinting, to the best of our knowledge.

- The face clustering technique which makes use of SIFT features evaluated on facial images.

The paper is organized as follows: in section 2, we introduce the face detection, facial feature extraction and face clustering methodology, the creation of the universal vocabulary and a procedure for incrementing the video database with new videos. Section 3 reviews the LDA framework. In section 4, we explain the training phase, as well as the query mechanism of the proposed fingerprinting framework. Its computational complexity is also analyzed in the same section. Experimental results are presented in Section 5. Finally, conclusions are drawn in section6.

2. Facial Feature Extraction and Data Organization

This section, outlines the facewords used in order to characterize a video and the proposed framework for video fingerprinting. For each video, two steps are undertaken:

a) Face detection. Faces are important semantic features for movies and humans often recognize a movie based on the actors that appear therein. Thus, we use actors' facial images as the basis of our video fingerprinting framework. These facial images are therefore interpreted adequately in facewords.

b) Face clustering using SIFT features. Since the proposed approach is based on face occurrences of specific actors in video frames, face detection is followed by

a face clustering step. This step is accomplished by evaluating facial image similarity based on SIFT features [31]. When applied on all videos in a video database, face clustering will ideally produce, in the training phase of the algorithm, the set of all actors appearing in the videos, i.e. one faceword per actor, as well as the number of facial image occurrences within each video. The face clustering approach is inspired from the work of Antonopoulos et al. [32].

The use of semantic information, namely actor appearances derived through face detection and clustering might seem to imply that the proposed framework inherits the current limitations and problems (false detection, erroneous matches etc) of these techniques. However, experimental results show that the effect of these limitations is rather small. This might be attributed to the fact that if these analysis modules err in a similar manner in both the query and the database videos, the influence of these errors in video fingerprinting performance is minimal.

2.1. Face Detection

The Viola and Jones face detector [33] is used in order to extract facial images from a video. We use the training set defined in OpenCV for frontal faces and, thus, the resulting facial images are frontal or nearly frontal ones. The fact that other facial poses, such as side views, are not detected does not affect the performance of the fingerprinting algorithm, since our aim is to characterize a video using face occurrences and not to achieve perfect pose-independent face detection. In other

words, as long as face detection performs in the same manner (e.g. produces frontal facial images only) in both the training videos and the query ones, there is no performance deterioration in video fingerprinting. Furthermore, if both frontal and side facial views were detected for the same person, it would be difficult for the clustering step to assign them to the same cluster. Some problems might arise when false detections (i.e. detections of regions that do not correspond to faces) occur. Such detections are few and temporally sparse enough. In addition, this issue is handled in the face clustering process that employs a threshold on the cardinality of each face cluster, thus, deleting small facial image clusters that most probably correspond to noise. In order to reduce processing time, face detection is performed every f video frames (typically $f = 10$). Obviously, infrequent face detection reduces both the processing time and the size of the facewords vocabulary for the database, since less faces are detected. Experiments involving different f values verified that this parameter indeed does not alter the results of the LDA inference, provided that f is kept below a reasonable threshold. For some video genres, like dramas, where long shots are the cinematographic rule, one can consider using larger values of f. In action movies, where shots alternate rapidly, small f values produce a more representative faceword vocabulary.

2.2. Face Clustering

The face clustering algorithm is based on the SIFT image features [31], which are robust local image descriptors associated with detected points of interest in an image. In our case, the SIFT image features are used to evaluate the similarity between facial images extracted from video frames through face detection. The adopted approach is straightforward. First, we calculate the SIFT interest points and the corresponding features on the facial images that resulted from the face detection. Subsequently, the technique described in [34] is used in order to find pairs of matching SIFT interest points across image pairs. The distance of the matched SIFT feature vectors is also evaluated. In order to declare that two faces belong to the same actor, two parameters are used: a) The Euclidean distance of the matched SIFT feature vectors and b) the total number of the matched SIFT points of interest. The two faces are considered to belong to the same actor, only if the number n_s of matched SIFT interest points is above a threshold T_n (which has been experimentally set to be equal to $T_n = 5$) and the arithmetic mean of distances $d_i \; i = 1, .., n_s$ of the SIFT feature vectors for the matched interest points, $\frac{1}{n_s} \sum_{i=1}^{n_s} d_i$ is below a certain threshold T. We have experimentally verified that this similarity evaluation approach gives good results even in cases where images produce few SIFT features, as is the case of facial images extracted from videos derived either from movies or short video clips, where the facial region of interest

(ROI) is relatively small (e.g., 80x80 pixels).

The face clustering approach, that aims to process all videos in the database in order to extract the global faceword vocabulary, proceeds as follows: The first face F_1 detected in a video is the seed for the first face cluster. Faces detected in subsequent frames are checked for similarity against F_1 using the similarity evaluation procedure described in the previous paragraph. If these facial images are found similar to this first facial image, they are assigned to the same cluster with F_1. If one of the subsequently detected facial images, e.g. face F_k, is found to be dissimilar to F_1, then a new face cluster, represented by F_k is created. Subsequently, facial images are tested against F_1 and F_k and, upon a positive match, are assigned to the corresponding face cluster, otherwise a new cluster is formed and so on. This procedure is applied to all videos in the database. Thus, if the same actor appears in several movies, this will create (ideally) only one face cluster, i.e. one faceword in the universal vocabulary. Clusters containing few entries are considered to be noise and, thus, are ignored.

A sample of the facial images assigned to 2 clusters (out of a total of 951 clusters) resulting from the application of the face clustering to the MUSCLE-VCD-2007 database [35] are depicted in Figure 1. At the end of the application of the face clustering procedure over the entire video database, the formed facial image clusters, represented by their first element/facial image (called facewords),

Figure 1: Sample facial images from two clusters created by the algorithm.

constitute the universal vocabulary $\mathcal{W}$ of this database having cardinality V. The cardinality of the universal vocabulary is equal to the number of the formed facial image clusters over the entire video database.

Based on the faceword vocabulary and the facial image associations to each faceword (cluster) we can create a faceword histogram for each video in the database. As will be shown later on, these histograms are used as an estimate of the probability distribution of the actors facial images in each video in the database, in order to estimate the model hyperparameters using an Expectation-Maximization (EM) algorithm. These histograms forms the rows of the so called faceword-by-video

matrix, in analogy to the term-by-document matrix in document modeling [16]. When a new video is to be added to the video database, face clustering/matching must be applied to only to the new video.

In more detail, the case of a new entry in the database is handled as follows: First face detection is performed on the new video and its facial images are matched to the universal faceword vocabulary. In case no new cluster is created (i.e. the actors in the new video are well represented by existing already by existing facewords) we do not update the universal vocabulary $\mathcal{W}$. If one or more new clusters are created (i.e. new actors appear in the new video), then we update the universal vocabulary, by adding the newly formed facewords. In both cases we need to run again the training phase of the LDA algorithm (see setion 4.1), so as to update its parameters and make the model accommodate the new video. In addition, one can defer the execution of the LDA training until a certain number of new videos have been gathered.

3. Latent Dirichlet Allocation

Many latent semantic analysis approaches have been proposed so far for multimedia analysis [36]. Latent Dirichlet Allocation (LDA) [16] is a recently proposed approach within this framework that produced good analysis and modeling results. In our case, we aim to use LDA to reveal the latent aspects of a video, based on actors appearances.As already briefly explained, LDA is a framework used until

now mainly in text retrieval and mining. LDA uses the following structures:

1. A finite universal vocabulary $\mathcal{W} = \{\mathbf{w}^1, \mathbf{w}^2, ..., \mathbf{w}^V\}$ of V words (i.e. basic units of discrete data). Each $\mathbf{w}^i$ with $i \in [1..V]$ is a vector where the i-th element is 1 and all others 0 (i.e. $\mathbf{w}^i(i) = 1$ and $\mathbf{w}^i(j) = 0$ for $i \neq j$). For simplicity we will refer to $\mathbf{w}^i(i)$ as w^i.
2. Documents where each document $\mathbf{v}$ is a sequence of N words from the universal vocabulary $\mathcal{W}$, $\mathbf{v} = (\mathbf{w}_1^{g(1)}, \mathbf{w}_2^{g(2)}, ..., \mathbf{w}_N^{g(N)})$, where g is a surjective map $g : [1..N] \rightarrow [1..V]$ and $\mathbf{w}_i^{g(i)}$ denotes that the i-th word in the sequence $\mathbf{v}$ is the $g(i)$-th word in the vocabulary $\mathcal{W}$. The fact that g is surjective, is because in $\mathbf{v}$, we can have multiple instances of the word $\mathbf{w}^i$.
3. A number of topics. The term topic is used to denote the latent vector variables $\mathbf{z}^i$, which represent probability distributions on sets of facewords. Topics are abstract notions which can not be assigned any physical meaning other than the one just mentioned.
4. Finally, a corpus $\mathcal{C}$ which represent a collection of documents.

In the proposed video fingerprinting framework, a word $\mathbf{w^i}$ is a faceword (i.e. a certain facial image, ideally corresponding to a particular actor) and each video $\mathbf{v}$ is a document. The universal vocabulary $\mathcal{W}$ is the set of all facewords, as discovered by the application of the face clustering procedure (the face clusters centers) on all videos of the database and ideally has cardinality equal to the number of actors

in the database.The term topic is used so as to provide an intuition that the latent aspects of a video, which are discovered under the bag-of-faces assumption may reveal distinction between the videos, based on a set of underline topics / themes. Finally, the corpus is equivalent to the video database. The motivation behind the adopted approach stems from the fact that the distribution of actor face appearances throughout a movie can provide a description of a video clip or a movie, which will be robust enough to be used for video fingerprinitng. Every video is described as a mixture of topic distributions and the mixture coefficients are used as the feature vector (i.e. the fingerprint) that characterize this video and is utilized in the matching/classification task typically involved in video fingerprinting.

The reason we have chosen the LDA model is because actor appearances may be considered as a multinomial experiment. In other words, if we have a set of actors, a movie can be thought as being constructed after several trials of “drawing” actors from a deck. Under such an assumption, the distribution of actors in a movie may be considered as a multinomial one. Moreover, we can assume that these multinomial distributions are not the same for all videos but they are parametrized by latent variables which are drawn from a Dirichlet distribution. The choice of the Dirichlet distribution in favor of others, is explained by the fact that the Dirichlet distribution is the conjugate prior of the multinomial distribution [19].

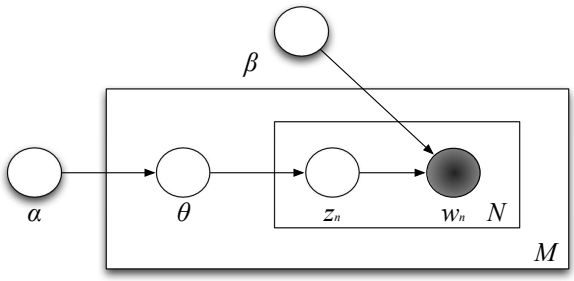

Figure 2: The LDA graphical model.

The graphical model of LDA is shown in Figure 2. Graphical models are used to represent conditional independence (exchangeability) among the random variables of a specific problem. In such a graph each node is a random variable, and the missing edges between nodes represent conditional independence (exchangeability). Hence, they provide means for the representation of the joint distribution between random variables [37]. In the LDA graphical model (Figure 2), the boxes represent replicates, that is, how many times a random variable needs to be sampled. Thus, the outer and inner boxes represent the selection of M videos that form the video database and the repeated choice (N times) of topics and facewords within a video, respectively.

In the rest of this section the basic equations of the LDA algorithm, adapted from [16] will be presented. A tutorial on this subject can be found in [38].

The LDA probabilistic model consists of the following generative process that creates a video $\mathbf{v}$ made up of a sequence of N facewords $(\mathbf{w}_1^{g(1)}, \mathbf{w}_2^{g(2)}, .., \mathbf{w}_N^{g(N)})$,

where each $\mathbf{w}_i^{g(i)}$ is drawn from a topic distribution [16]:

- Choose N from a Poisson distribution $Poisson(\xi)$
- Choose a K-dimensional random vector variable $\boldsymbol{\theta}$= $[\theta_1, \theta_2, ..., \theta_K]^T$ from a Dirichlet distribution: $\boldsymbol{\theta} \sim Dir(\boldsymbol{\alpha})$, where $\boldsymbol{\alpha}$ is the vector hyperparameter of the prior Dirichlet distribution.
- For each of the N facewords $\mathbf{w}_n^{g(n)}$:
  - Choose a topic $\mathbf{z}_n^{h(n)}$ from a multinomial distribution [39] parametrized with $\boldsymbol{\theta}$. $\mathbf{z}_n^{h(n)} \sim Multinomial(\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is a Dirichlet distributed vector variable and h a surjective map $h : [1, N] \rightarrow [1, K]$ which provides that the n-th word is conditioned from the $h(n)$-th topic in the latent topics set of cardinality K. The function h is surjective because in $\mathbf{v}$, we can have multiple instances of the same topic $\mathbf{z}^i$.
  - Choose a word $\mathbf{w}_n^{g(n)}$ from $p(\mathbf{w}_n^{g(n)}|\mathbf{z}_n^{h(n)}, \boldsymbol{\beta})$, which will also be a multinomial distribution.

The fact that N (number of facewords in each video) is a Poisson random variable is not critical for the algorithm. The above generative process, suggests that each faceword is generated with a probability conditioned on a topic (the latent variable). The topics, in turn, are generated from a multinomial distribution with a Dirichlet

prior (i.e. $\boldsymbol{\theta}$), which is an assumption based on the fact that the Dirichlet distribution is a conjugate prior to the multinomial distribution and thus the most natural choice for a prior [19]. The dimensionality K of the multinomial latent variable $\mathbf{z}_n^{h(n)}$ can not be known a priori, and furthermore, no methods for its estimation exist until now. In general, defining the dimensionality of the latent variable in the LDA model is still an open issue and certainly beyond the scope of this paper.

Let us suppose that we fix the dimensionality of the topic variable to K, and thus, the latent set of topics $\mathcal{Z}$ contains K distinct topics $\mathcal{Z} = \{\mathbf{z}^1, \mathbf{z}^2, ..., \mathbf{z}^K\}$ where $\mathbf{z}^i$ is a vector where the i-th element is 1 and all others 0 (i.e. $\mathbf{z}^i(i) = 1$ and $\mathbf{z}^i(j) = 0$ for $i \neq j$). For simplicity $\mathbf{z}^i(i)$ will be denoted as z^i. The $K-$dimensional Dirichlet random vector variable $\boldsymbol{\theta}$, which represents the mixture of topics distributions in a video, is chosen from a distribution with probability density function:

$$p(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \frac{\Gamma(\sum_{i=1}^{K} \alpha_i)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} \cdots \theta_K^{\alpha_K - 1}, \tag{1}$$

where $\boldsymbol{\theta}$ lies in a $(K-1)$-simplex (due to the fact that $\theta_i \geq 0$ and $\sum_{i=1}^{K} \theta_i = 1$), $\boldsymbol{\alpha}$ is the K-dimensional Dirichlet vector hyperparameter with $\alpha_i > 0$ and $\Gamma(x)$ is the Gamma function.

The $K \times N$ parameter matrix $\boldsymbol{\beta}$ contains the probabilities $\beta(i, j)$ that the face-word $\mathbf{w}^j$ is generated from topic $\mathbf{z}^i$. The parameter matrix $\boldsymbol{\beta}$ is not known and has to be estimated, as we shall demonstrate later on, from a variational EM algorithm.

Given the hyperparameter $\boldsymbol{\alpha}$ and the matrix parameter $\boldsymbol{\beta}$, we can calculate the joint distribution of a topic mixture $\boldsymbol{\theta}$, a set of N topics $\mathbf{Z} = (\mathbf{z}_1^{h(1)}, \mathbf{z}_2^{h(2)}, ..., \mathbf{z}_N^{h(N)})$ and a video $\mathbf{v}$ (sequence of N words) by:

$$p(\boldsymbol{\theta}, \mathbf{Z}, \mathbf{v}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = p(\boldsymbol{\theta}|\boldsymbol{\alpha}) \prod_{n=1}^{N} p(\mathbf{z}_n^{h(n)}|\boldsymbol{\theta}) p(\mathbf{w}_n^{g(n)}|\mathbf{z}_n^{h(n)}, \boldsymbol{\beta}). \quad (2)$$

The above formula is a straightforward application of the relations schematically depicted in the graphical model in Figure 2 and of the probabilities product rule. In this formula, $p(\mathbf{z}_n^{h(n)}|\boldsymbol{\theta})$ is the probability that the topic $\mathbf{z}_n^{h(n)}$ is chosen with a Dirichlet prior and equals the $h(n)$-th element $\theta_{h(n)}$ of the vector $\boldsymbol{\theta}$. By integrating (2) over $\boldsymbol{\theta}$ and summing over $\mathbf{z}_n^{h(n)}$, we obtain the marginal distribution for a video $\mathbf{v}$:

$$p(\mathbf{v}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \int p(\boldsymbol{\theta}|\boldsymbol{\alpha}) \left(\prod_{n=1}^{N} \sum_{n=1}^{N} p(\mathbf{z}_n^{h(n)}|\boldsymbol{\theta}) p(\mathbf{w}_n^{g(n)}|\mathbf{z}_n^{h(n)}, \boldsymbol{\beta}) \right) d\boldsymbol{\theta}. \quad (3)$$

4. Video Fingerprinting Using LDA

4.1. Training Through Variational Inference

In order to train our model we introduce into the LDA algorithm the histograms of facewords for each video, which are produced by the face detection/clustering procedure. The faceword-by-video matrix created from this process is considered as the first estimate of the matrix $\boldsymbol{\beta}$ in the LDA framework. Training the model involves in fact solving the inference problem of computing the posterior distribution of the hidden vectors $\boldsymbol{\theta}$ and $\mathbf{z}_n$ given a video $\mathbf{v}$ and the Dirichlet parameters $\boldsymbol{\alpha}$

and $\boldsymbol{\beta}$:

$$p(\boldsymbol{\theta}, \mathbf{Z}|\mathbf{v}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{p(\boldsymbol{\theta}, \mathbf{Z}, \mathbf{v}|\boldsymbol{\alpha}, \boldsymbol{\beta})}{p(\mathbf{v}|\boldsymbol{\alpha}, \boldsymbol{\beta})}. \quad (4)$$

Unfortunately, the computation of this distribution is in general intractable due to $p(\mathbf{v}|\boldsymbol{\alpha}, \boldsymbol{\beta})$. Indeed by substituting (1) in (4) we obtain:

$$p(\mathbf{v}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \int \left(\prod_{i=1}^{K} \theta_i^{\alpha_i - 1} \right) \left(\prod_{n=1}^{N} \sum_{i=1}^{K} \prod_{j=1}^{V} (\theta_i \cdot \beta(i,j))^{w_n^j} \right) d\boldsymbol{\theta}. \quad (5)$$

The integral in (5) is intractable, due to the coupling between $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ [40]. However, a wide variety of approximate inference algorithms can be used to this end, including Laplace approximation, variational approximation, and several Markov chain Monte Carlo methods [41]. In our case, we use a variational inference method as proposed in [16]. The idea, as in all variational methods, is to use Jensen's inequality [41] and find a lower bound of the log likelihood function [41]. To do so, a family of lower bounds, indexed by a set of variational parameters is considered. The variational parameters are selected through an optimization procedure that attempts to find the tightest possible lower bound. For this purpose, one introduces a Bayesian network graph (Figure 3), where the edges between $\boldsymbol{\theta}$, $\mathbf{z}_n$ and $\mathbf{w}_n$ are dropped to resolve the coupling of $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$, which makes (5) intractable. Two variational parameters $\boldsymbol{\phi}$ and $\boldsymbol{\gamma}$ are inserted and, thus, we obtain a family of distributions of the latent variables:

$$q(\boldsymbol{\theta}, \mathbf{Z}|\boldsymbol{\gamma}, \boldsymbol{\phi}) = q(\boldsymbol{\theta}|\boldsymbol{\gamma}) \prod_{n=1}^{N} q(z_n, \boldsymbol{\phi}_n), \quad (6)$$

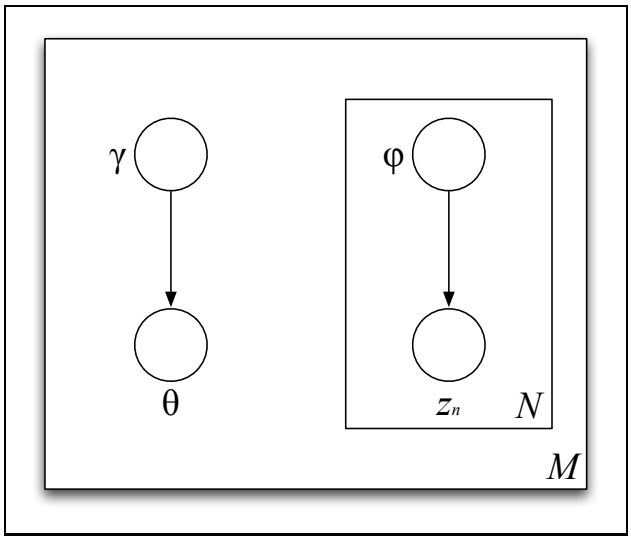

Figure 3: Variational Inference Model.

where $\boldsymbol{\gamma}$ is a K-dimensional Dirichlet distributed parameter vector and $(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, .., \boldsymbol{\phi}_N)$ are vectors of multinomial distributed parameters. In [16], it is proven that the values $\boldsymbol{\gamma}^*$ and $\boldsymbol{\phi}^*$ that lead to a tight lower bound on the log likelihood can be evaluated through the following optimization problem:

$$(\boldsymbol{\gamma}^*, \boldsymbol{\phi}^*) = \arg \min_{(\boldsymbol{\gamma}, \boldsymbol{\phi})} KL(q(\boldsymbol{\theta}, \mathbf{Z}|\boldsymbol{\gamma}, \boldsymbol{\phi}) \| p(\boldsymbol{\theta}, \mathbf{Z}, \mathbf{v}|\boldsymbol{\alpha}, \boldsymbol{\beta})), \tag{7}$$

where KL is the Kullback-Leibler divergence [39]. The optimization procedure is described in [16]. By taking the derivatives of (7) we have the following update functions for $\boldsymbol{\phi}$ and $\boldsymbol{\gamma}$:

$$\phi_n^i = \beta(i, \arg_j\{\mathbf{w}_n^j = 1\}) \exp(E_q[log(\theta_i)|\gamma]), \tag{8}$$

$$\gamma_i = \alpha_i + \sum_{n=1}^{N} \phi_n^i, \tag{9}$$

where ϕ_n^i is the probability that the n-th faceword is derived from the i-th topic.

We note that $\boldsymbol{\gamma}^*$ is a function of $\mathbf{v}$ due to the fact that (7) is evaluated for fixed $\mathbf{v}$, and thus, provides a unique representation of a video from the training set, in the simplex formed from the topics. In other words, each training video is uniquely characterized as a point in this $(K-1)$-simplex.

The parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, involved in the model, are estimated by training our model. To do so, we follow the approach in [16] which is an empirical Bayes method and consists of the following EM process:

- (E-step): For each video, find the optimal values of the variational parameters $\boldsymbol{\phi}^*$, $\boldsymbol{\gamma}^*$. This estimation step uses the aforementioned methodology for fixed values of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$

- (M-step): Maximize the resulting lower bound on the likelihood with respect to parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, using (8) and (9) solved for $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$, respectively.

At the end of the training procedure, apart from the values of $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, the parameter vector $\boldsymbol{\gamma}$ (called video fingerprint) for each of the database videos, which characterizes this video, is evaluated.

4.2. Querying the database

Assuming that the parameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ have been estimated from the training video set, we have to develop a method for finding if a query video is a replica

of a training video or not. The video is first subjected to face detection and, then, each of the detected facial images is mathced to one of the universal vocabulary facewords. Thus, the query video is represented as a sequence of words $\mathbf{v}_{query} = (\mathbf{w}_1^{g(1)}, \mathbf{w}_2^{g(2)}, .., \mathbf{w}_N^{g(N)})$. The query video is then characterized by the K-dimensional parameter $\boldsymbol{\gamma}^*$ which is an estimate of the mixture of topics distributions $p(\boldsymbol{\theta}, \mathbf{Z}|\mathbf{v}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ in this video and is found via inference with the trained model using (7). Thus $\boldsymbol{\gamma}^*$ is used as the feature vector (i.e., fingerprint) of the query video.

In order to decide whether $\mathbf{v}_{query}$ is a replica of one of the videos in the database, we use the KL divergence between its variational parameter $\boldsymbol{\gamma}^*$ and the ones of the videos stored in the database. By doing so, we find the index F of the closest database video:

$$F = \arg\min_i KL(\boldsymbol{\gamma}^*(\mathbf{v}_i) \| \boldsymbol{\gamma}^*(\mathbf{v}_{query})), \tag{10}$$

where $\mathbf{v}_i$ is i-th video in the database. Therefore, $\mathbf{v}_F$ is the database video that is closest to $\mathbf{v}_{query}$ and thus a candidate for being the corresponding original video.

Besides matching query videos to the ones in the video database, we also need to handle videos that are not replicas of the ones in the database. This is done in a two-level process. First, query videos whose facial images do not match any (or match only few) facial images stored in the database (universal) vocabulary are characterized as not matching with any of the videos in the database. In case

a video has enough face matches with the database vocabulary (typically more than 20 matches) we decide that the query video $\mathbf{v}_{query}$ matches video $\mathbf{v}_F$ in the database only if the KL divergence in (10) is below a certain threshold T_{KL}. This threshold is experimentally found by introducing into the system query videos that are not in the database but have enough face matches with the database as well as videos taken from within the database and evaluating the threshold that minimizes the false acceptance and false rejection ratios. This threshold was found to be equal to $T_{KL} = 0.95$.

5. Experimental Results

As mentioned before, the performance of the method has been evaluated on three video data sets namely Video Clips ($\mathcal{VC}$), Movies ($\mathcal{M}$) and the MUSCLE-VCD-2007 database ($\mathcal{M} - \mathcal{VCD}$). The $\mathcal{VC}$ data set includes short, low quality videos, randomly collected over the Internet (YouTube). It consists of 332 videos, each 2-5 minutes long (approximately 4000-7000 frames per video clip). The $\mathcal{M}$ data set consists of 8 high-quality full length movies of approximately 2 hours each (approximately 150.000 video frames). The small number of movies in $\mathcal{M}$ was due to copyright issues. Finally, the $\mathcal{M} - \mathcal{VCD}$ database is a new fingerprinting / copy detection benchmark database [35], which consists of about 100 hours of video material coming from different sources: web video clips, TV archives and movies. The videos have different bitrates, different resolutions, different video

formats and cover a very large range of genres: documentaries, movies, sports events, TV shows, cartoons etc. The three databases were selected in order to test three different cases that can be encountered. A database with a large number of short videos (YouTube), a movie databases and, finally, a mixture of long and short length videos in a database of a total of 100 hours of videos.

In all video data sets, we have applied face detection every 10 frames ($f = 10$). Even at this, relatively low, face detection frequency (one face detection every approximately 0.5 sec) we are almost sure that we will detect all actor faces involved in the video. For the $\mathcal{VC}$ and $\mathcal{M} - \mathcal{VCD}$ data sets, the length of the universal vocabulary was 1088 and 951 respectively (i.e. 1088 and 951 different facewords, representing facial image clusters, were created, respectively). Of course, this number does not correspond to the actual total number of actors that appear in these videos, due to errors introduced by the face detector (detection of image regions that do not correspond to faces) and the face clustering algorithm (e.g. two or more clusters corresponding to the same actor might be created). One can think that these errors may introduce fingerprinting errors. However, as long as the face detection and clustering methods perform consistently, i.e., in a similar manner, in the query and database videos, these errors do not cause serious problems to the fingerprinting task.

In order to evaluate the performance of the proposed method, three different

Table 1: Fingerprinting performance metrics for the three benchmarking video sets.

	TEST A		TEST B
	MC	FR	FA
$\mathcal{VC}$	2.11%	1.2%	0%
$\mathcal{M}$	0%	0%	0%
$\mathcal{M}-\mathcal{VCD}$	5.19%	0%	0%

types of experiments have been performed:

1. Tests with query videos that are replicas of the database videos (test A). Two types of errors are expected in this case: a misclassification error (MC), measured by the percentage of query videos that were classified to a wrong original video in the database and the false rejection error (FR), which is the percentage of query videos that were erroneously tagged as not belonging to the database.
2. Tests with query videos that do not belong to the database (test B). In this case, the performance is measured in terms of the false acceptance (FA) error, i.e., the percentage of query videos that are erroneously tagged as being a replica of a database video.
3. For the $\mathcal{M}-\mathcal{VCD}$ database, we have also implemented the experiments described in [35]. According to the evaluation protocol, specific query videos that are provided with the database and are transformed versions of the database videos, as well as videos that do not belong to the database are used.

These experiments have been conducted in order to compare our method with other methods that have also been tested on the same data set with the same testing protocol.

The first set of experiments aimed at showing the system performance when queried with videos that are identical with those in the database i.e., they have not been manipulated. In test A, the $\mathcal{VC}$ data set involved 332 videos which were used to both populate the system database (and train the system) and as query videos. In data set $\mathcal{M}$, 8 videos were used. Finally, in the $\mathcal{M}-\mathcal{VCD}$ data set, 77 videos were used. For test B, in the $\mathcal{VC}$ data set, we trained the model and populated the system database with 247 videos out of the original 332 and used the remaining 85 videos as query videos for testing. For data set $\mathcal{M}$, we populated the training database with 6 videos and used 2 movies as query videos. Similarly, for $\mathcal{M}-\mathcal{VCD}$ we placed 58 videos in the training database and used 19 as queries. Results are depicted in Table 1

As can be seen, the MC and FR errors for the $\mathcal{VC}$ and $\mathcal{M}-\mathcal{VCD}$ database are sufficiently low, whereas all other errors, including all errors for the $\mathcal{M}$ data set, are zero. The obtained performance metrics suggest that the fingerprinting performance of the proposed framework scales well enough: by increasing the database size five times (from the $\mathcal{VC}$ to the $\mathcal{M}-\mathcal{VCD}$ database) the MC error doubles, whereas the FR error vanishes. It should be noted here that the LDA

model was initially introduced for document modeling, where the number of terms and documents are sometimes in the order of millions. This fact suggests that the discriminative power of the LDA model can scale to large databases.

The false acceptance (FA) rate can be further analyzed due to the fact that, as already mentioned in the previous section, declaring that a video is not a replica of a video in the database is a two step process, as described in Section 2.2. In our experiments (test B), out of the 85 non-replica videos that were used for querying the $\mathcal{VC}$ data set, only 4 of them (that is 4.7%) were rejected at the first step, i.e. due to the small number of face matches with the vocabulary. In the $\mathcal{M}$ database, however, both query videos were rejected in the first step. This can be explained by the fact that, in high-quality movies, the face detector and the face clustering algorithm perform far better than in the low-quality internet videos, thus resulting in a better discrimination from the first step of matching/rejection. In the $\mathcal{M} - \mathcal{VCD}$, no videos were rejected at the first step.

It should be noticed that, for the test A, we have conducted experiments for different K values (number of topics). Results for the misclassification error, as a function of K are presented in Figures 4 and 5. For $\mathcal{VC}$ the best results were achieved for $K = 329$ (MC error 2.11% and FR error 1.20%) and for $\mathcal{M}$ the best results were achieved for $K > 9$ with an MC error 0% and FR Error 0%. Finally, for $\mathcal{M} - \mathcal{VCD}$ the best results were achieved for $K = 120$ (MC error

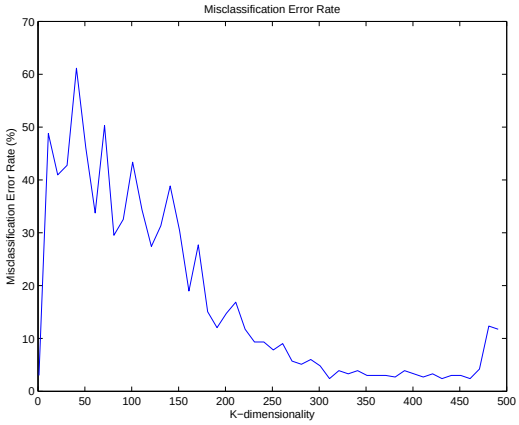

Figure 4: Misclassification error for $\mathcal{VC}$ versus the number of topics K.

5.19% and FR error 0%). As mentioned before, the selection of the number of topics K that provides the best results is still an open issue for LDA. From our experiments it seems that K values close to the number of videos in the database are good choices, for this specific application of LDA. Experiments with varying K should be performed to establish a good K value for each new training video database.

Another set of experiments aimed to test our framework against frequently encountered attacks. Thus, we have performed tests for five types of video processing operations (attacks):

- Histogram Equalization (HE)
- 10% Temporal Cropping (TC)

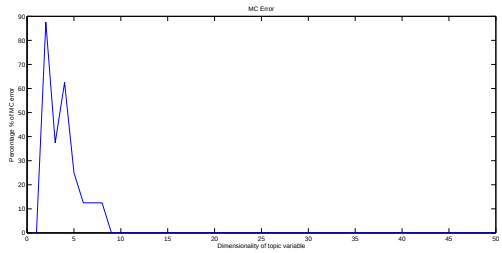

Figure 5: Misclassification error for the feature length movies data set $\mathcal{M}$ versus versus the number of topics K.

- 10% Spatial Frame Cropping (removal of left-right frame columns) (SFC)
- 10% Frame Removal (removal of randomly selected frames) (FrR)
- Spatial Subsampling (downsizing) by a factor of 2 (S)
- Video Compression (C) 50% bit-rate reduction

We applied the attacks on the query videos of the $\mathcal{VC}$ data set and performed both tests A, B. For the HE, TC, SFC and FrR attacks the performance metrics remained unaltered (MC=2.11%, FR=1.2%, FA=0%), due to the fact that both the face detector and the face clustering algorithm provided practically the same results on the attacked video clips as on the original ones. However, in the spatial subsampling attack (S), the face detector produced poor results, due to the fact that the original video clips resolution was 320×240 pixels and dropped down to 160×120 pixels, a resolution that is very small for correct face detection. In this case, the face detector could detect faces in only 154 of the 332 total query videos,

and, therefore, the remaining 178 videos were automatically declared as not matching to any of the database videos, increasing dramatically the fingerprinting error (MC, FR and FA) to more than 40%.

It should be noted that our framework is rather vulnerable to extensive temporal cropping, due to the fact that the proposed framework uses global information (faceword histograms). It is obvious that if, due to temporal cropping, the query video depicts a small subset of the actors appearing in the original video, the framework will fail to recognize it as a replica.

For the video compression attack, we used the $\mathcal{M}$ database. We have compressed each video with standard lossy compression schemes (DivX, Xvid and others) to 50% of the original bitrate. Results were not altered, due to the fact that the face detector and clustering performed correctly. Moreover, we where expecting this behavior due to experiments performed in the $\mathcal{M} - \mathcal{VCD}$ database, which contains query videos with compression attacks. We did not perform such attacks to the $\mathcal{VC}$ database, due to the fact that the videos therein were low-quality, already compressed ones, gathered from websites. Thus, any further compression would degrade them so they become useless for testing.

Finally, as previously mentioned in this section, we have conducted experiments on the $\mathcal{M} - \mathcal{VCD}$ database using the ST1 query set (15 videos) defined in the evaluation protocol described in [35]. The videos in this query set have been

modified by combinations of rather sever image processing manipulations, such as color change and blurring, re-encoding, cropping and color change, recording the video with a camcorder and subtitles addition etc. The set ST1 includes also videos that do not belong to the database. The performance metric used in this case was the so-called ST1 score, namely the ratio of correct answers to the number of queries. In our case, 24 out of the 101 videos of the data set were not included in the database, since they did not generate a sufficient number of detected faces. In addition, out of the 10 query videos that were replicas of videos in the database, 4 were replicas of the 24 videos that were excluded from the training database. Thus their ground-truth tags had to be changed from "replica of database video x" to "not existing in the database". Finally, 3 out of the 15 query videos were excluded from the test videos, since they did not provide sufficient face detections. Due to these necessary modifications to the experiment, the ST1 score of 0.83 (10 out of 12 videos were recognized correctly) achieved by our method is not directly comparable to the scores achieved by other fingerprinting methods tested on [35]. Nevertheless, this score is a very good indication that our method can achieve results beyond the state of the art, since the second best ST1 score in this experimental setup was 0.8. In terms of execution time, our method required 87 minutes to process the 12 query videos. Four teams (some with more than one approaches) have tested their algorithms on this database and query set (see [35]) and reported ST1

scores ranging from 0.46 (worse) to 0.86 (best) and query execution times from 14 minutes to 99 minutes. The method that obtained the best score is described in [42].

All the experiments have been conducted on a Hewlett-Packard HP xw6600 workstation with Intel® Xeon® CPU at 2.50GHz and 3.25GB of RAM. The time spent for the training phase (database construction) in each experiment was approximately 12 hours, 5 hours and 47 hours on the three viedeo sets respectively. Thus, in the worst case, the training phase takes a bit more than half the duration of the videos.

The testing (query) phase of the video fingerprinting framework consists of first applying face detection to the query, an operation that obviously has the same time complexity as the one used in the training phase. Next, the universal vocabulary facewords, created in the training phase are matched to the facial images produced by the face detector, which is a relatively time consuming process. For a query video of approximately 1 hour duration, more than 24 minutes are needed to create the faceword histogram for a universal facewords vocabulary of $V = 951$ facewords in total. This time depends mainly on the size of the universal facewords vocabulary cardinality V. The LDA inference step is fast and requires less than 10 seconds. Finally, comparing the resulting feature vector produced by the LDA inference with the ones stored in the database takes less than 1 second in all video

sets. This time increases linearly with the number of videos in the database and is independent from the video duration.

From the complexity characteristics of the proposed method presented above, we can conclude that the time required for training is linearly related to the length of the videos in the database. The experimental results presented above i.e., that the training is at worst twice as fast as realtime, verify this conclusion. The query phase complexity is also linearly related to the length of the video database. These properties ensure that the framework scales well in terms of execution time with the database size.

6. Discussion, Conclusions and Future Work

In this work, a new framework for video fingerprinting has been presented. The intuition behind this work is that actor instances (i.e., mapped to facewords) can carry a significant amount of information and can be used to capture very distinctive video features, thus characterizing uniquely each video. In this context, by applying a generative probabilistic model, namely the Latent Dirichlet Allocation, we aim at discovering latent aspects of a video (video topics), based on the semantic information related to actor appearances. The probability distribution of these video topics, for each video, can be used efficiently for video fingerprinting applications.

The experimental results provided in this paper show that our effort to adapt a

language modeling technique to video fingerprinting, although not straightforward, can be quit efficient, as proven by our experiments. In addition, the proposed method has good scalability with respect to the database size, in terms of both fingerprinting performance and computational effort.

The proposed framework is based on actor appearances since these provide distinctive semantic information, and because humans appear in most videos genres (movies, news, TV series etc.). The proposed framework is not directly applicable to video content where human, appearances are rare or non-existent such as certain documentaries, e.g., those dealing with animals. However, this framework can be easily extended to such cases as well, since "actors" can be animals, even objects and scene artifact.

In the future, we shall further explore the proposed approach, by using a more complex vocabulary that will include, for example, human pose, human interactions etc. By doing so we hope to provide a better representation of a video topics, and, thus, a more robust and discriminative fingerprinting algorithm.

References

[1] D. Kundur, K. Karthik, Video Fingerprinting and Encryption Principles for Digital Rights Management, Proceedings of the IEEE 92 (6) (2004) 918–932.

[2] C. De Roover, C. De Vleeschouwer, F. Lefebvre, B. Macq, Robust video

hashing based on radial projections of key frames, IEEE Transactions on Signal Processing 53 (10 Part 2) (2005) 4020–4037.

[3] A. Kołcz, J. Alspector, Improved robustness of signature-based near-replica detection via lexicon randomization, in Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining (2004) 605–610.

[4] M. Mıhçak, R. Venkatesan, New iterative geometric methods for robust perceptual image hashing, Security and privacy in digital rights management (2002) 13–21.

[5] J. Seo, J. Haitsma, T. Kalker, C. Yoo, A robust image fingerprinting system using the Radon transform, Signal Processing: Image Communication 19 (4) (2004) 325–339.

[6] S. Kozat, R. Venkatesan, M. Mihçak, Robust perceptual image hashing via matrix invariants, in: International Conference on Image Processing, Vol. 5, IEEE, 2004, pp. 3443–3446.

[7] S. Lin, M. Ozsu, V. Oria, R. Ng, An extendible hash for multi-precision similarity querying of image databases, in: Proceedings Of The International Conference On Very Large Data Bases, Citeseer, 2001, pp. 221–230.

[8] V. Monga, B. Evans, Robust perceptual image hashing using feature points, in: Image Processing, 2004. ICIP'04. 2004 International Conference on, Vol. 1, IEEE, 2004, pp. 677–680.

[9] P. Indyk, G. Iyengar, N. Shivakumar, Finding pirated video sequences on the internet, Tecnical Report, Stanford University.

[10] J. Oostveen, T. Kalker, J. Haitsma, Feature extraction and a database strategy for video fingerprinting, Proceedings of the 5th International Conference on Recent Advances in Visual Information Systems VISUAL '02 (2002) 117–128.

[11] B. Coskun, B. Sankur, N. Memon, Spatio–temporal transform based video hashing, Multimedia, IEEE Transactions on 8 (6) (2006) 1190–1208.

[12] A. Hampapur, R. Bolle, Comparison of sequence matching tecniques for video copy detection, Conference on Storage and Retrieval for Media Databases (2002) 194–201.

[13] J. Law-To, V. Gouet-Brunet, O. Buisson, N. Boujemaa, Video Copy Detection on the Internet: The Challenges of Copyright and Multiplicity, IEEE International Conference on Multimedia and Expo (2007) 2082–2085.

[14] K. Changick, B. Vasudev, Spatiotemporal sequence matching for efficient

video copy detection, IEEE Transactions on Circuits and Systems for Video Technology 15 (2005) 127–132.

[15] S. Lee, C. D. Yoo, Robust video fingerprinting based on 2D-OPCA of affine covariant regions, in: Proceedings of IEEE International Conference on Image Processing, 2008.

[16] D. Blei, A. Ng, M. Jordan, Latent dirichlet allocation, The Journal of Machine Learning Research 3 (2003) 993–1022.

[17] B. De Finetti, B. de Finetti, Theory of Probability, volume I, Bull. Amer. Math. Soc. 83 (1977), 94-97. DOI: 10.1090/S0002-9904-1977-14188-8 PII: S 2 (9904) (1977) 14188–8.

[18] X. Wei, W. Croft, LDA-based document models for ad-hoc retrieval, Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval (2006) 178–185.

[19] L. Fei-Fei, P. Perona, A bayesian hierarchical model for learning natural scene categories, in: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, Vol. 2, Ieee, 2005, pp. 524–531.

[20] F. Monay, D. Gatica-Perez, On image auto-annotation with latent space mod-

els, Proceedings of the eleventh ACM International Conference on Multimedia (2003) 275–278.

[21] J. Sivic, B. Russell, A. Efros, A. Zisserman, W. Freeman, Discovering object categories in image collections, Proceeding of International Conference on Computer Vision 1 (2005) 65–69.

[22] B. Russell, W. Freeman, A. Efros, J. Sivic, A. Zisserman, Using multiple segmentations to discover objects and their extent in image collections, in: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, Vol. 2, IEEE, 2006, pp. 1605–1614.

[23] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. Blei, M. Jordan, J. Kandola, T. Hofmann, T. Poggio, J. Shawe-Taylor, Matching Words and Pictures, Journal of Machine Learning Research 3 (6) (2003) 1107–1135.

[24] R. Fergus, L. Fei-Fei, P. Perona, A. Zisserman, Learning Object Categories from Google's Image Search, in: Proceedings of the Tenth IEEE International Conference on Computer Vision, Vol. 2, 2005.

[25] J. Niebles, H. Wang, L. Fei-Fei, Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words, International Journal of Computer Vision (2008) 1–20.

[26] J. Cao, J. Li, Y. Zhang, S. Tang, LDA-Based Retrieval Framework for Semantic News Video Retrieval, Proceedings of the International Conference on Semantic Computing (2007) 155–160.

[27] J. Yang, Y. Liu, E. Xing, A. Hauptmann, Harmonium-based models for semantic video representation and classification, in: Proceedings of the Seventh SIAM International Conference on Data Mining, 2007, pp. 378–389.

[28] M. Héritier, S. Foucher, L. Gagnon, Key-places detection and clustering in movies using latent aspects, in: IEEE International Conference on Image Processing, Vol. 2, IEEE, 2007, pp. 221–225.

[29] D. Larlus, F. Jurie, Latent mixture vocabularies for object categorization, in: British machine vision conference, Citeseer, 2006, pp. 959–968.

[30] M. Fleischman, Unsupervised content-based indexing of sports video, Proceedings of the international workshop on Workshop on multimedia information retrieval (2007) 87–94.

[31] D. Lowe, Object recognition from local scale-invariant features, Proceedings of International Conference on Computer Vision 2 (1999) 1150–1157.

[32] P. Antonopoulos, N. Nikolaidis, I. Pitas, Hierarchical Face Clustering using

SIFT Image Features, Computational Intelligence in Image and Signal Processing, 2007. CIISP 2007. IEEE Symposium on (2007) 325–329.

[33] P. Viola, M. Jones, Robust real-time object detection, International Journal of Computer Vision 57 (2) (2002) 137–154.

[34] Q. Wang, S. You, Fast Similarity Search for High-Dimensional Dataset, Proceedings of the Eighth IEEE International Symposium on Multimedia (2006) 799–804.

[35] J. Law-To, A. Joly, N. Boujemaa, Muscle-vcd-2007: a live benchmark for video copy detection, http://www-rocq.inria.fr/imedia/civr-bench/ (2007).

[36] C. Snoek, M. Worring, Multimodal Video Indexing: A Review of the State-of-the-art, Multimedia Tools and Applications 25 (1) (2005) 5–35.

[37] M. Jordan, Learning in Graphical Models, Kluwer Academic Publishers, 1998.

[38] G. Heinrich, Parameter estimation for text analysis, Web: http://www. arbylon. net/publications/text-est. pdf.

[39] A. Papoulis, S. Pillai, Probability, random variables, and stochastic processes, McGraw-Hill New York, 1991.

[40] J. Dickey, Multiple hypergeometric functions: Probabilistic interpretations and statistical uses, Journal of the American Statistical Association 78 (383) (1983) 628–637.

[41] M. Jordan, Z. Ghahramani, T. Jaakkola, L. Saul, An Introduction to Variational Methods for Graphical Models, Machine Learning 37 (2) (1999) 183–233.

[42] S. Poullot, O. Buisson, Scalable mining of large video databases using copy detection, ACM New York, NY, USA, 2008.