Scalable Face Track Retrieval in Video Archives using Bag-of-Faces Sparse Representation

Bor-Chun Chen, Yan-Ying Chen, Yin-Hsi Kuo, Thanh Duc Ngo, Duy-Dinh Le, Shin'ichi Satoh, Winston H. Hsu

***Abstract*—Huge video archives consisting of news programs, dramas, movies, and web videos (e.g., YouTube) are available in our daily life. In all these videos, human is usually one of the most important subjects. Using state-of-the-art techniques, we can efficiently detect and track faces in the videos. In order to organize large-scale face tracks, containing sequences of (detected) consecutive faces in the videos, we propose an efficient method to retrieve human face tracks using bag-of-faces sparse representation. Using the proposed method, a face track is encoded as a single bag-of-faces sparse representation and therefore allowing efficient indexing method to handle large-scale data. To further consider the possible variations in face tracks, we generalize our method to find multiple sparse representations, in an unsupervised manner, to represent a bag of faces and balance the trade-off between performance and retrieval time. Experimental results on two real-world (million-scale) datasets confirm that the proposed methods achieve significant performance gains compared to different state-of-the-art methods.**

***Index Terms*—Face Track Retrieval, Bag-of-Faces Sparse Representation, Multiple Sparse Representations**

I. Introduction

Huge collections of videos are generated everyday in the form of news program, drama, movies, web videos, family recordings, etc. How to efficiently manage and mine information from these videos is a really important topic for many researchers. In all of these videos, human is usually one of the most important subjects; therefore, many studies focus on manipulating human faces (i.e., retrieval, recognition, annotation, etc.) in the videos [1], [2], [3], [4].

Different from traditional face recognition in still images, face recognition in videos can benefit from additional temporal redundancy because faces detected from consecutive frames at the similar location are usually of the same person. Using this extra information, face recognition based on sets of images is applied to improve the accuracy. Such face sequences detected from the videos can be regarded as a *face track* or *bag of faces*.

With the explosive growth of the videos, besides of face recognition, the emerging research is to conduct content-based face track retrieval [5], [6]. However, most of the existing face recognition methods for image set rely on complex distance measures between two sets of faces and therefore can not easily work with current index frameworks, which are essential as witnessing the exponential growth of the video collections.

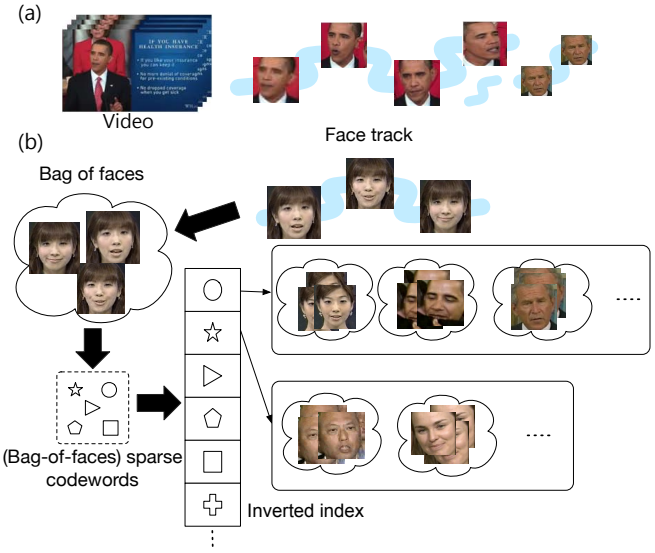

Fig. 1. Illustration of the proposed method. (a) The sheer amount of videos is available nowadays and millions of faces can be detected and tracked in the videos. (b) We aim to efficiently retrieve face tracks extracted from videos as the query and the target large-scale collections. In our work, each face track is represented by a bag-of-faces sparse representation – exploiting temporal redundancy in the videos. Non-zero entries of the sparse representation are then used as codewords for building inverted index and enabling scalable and effective retrieval in large-scale data.

To overcome this problem, we propose a novel coding method to encode the bag of faces into a single sparse representation. As shown in Figure 1, each bag of faces is represented by a sparse representation, using the non-zero entries in the sparse representation as discrete codewords, inverted index is built with millions of faces extracted from videos and can enable scalable retrieval over large-scale database. To improve the retrieval performance, we further generalize the proposed coding method to find multiple sparse representations which might accommodate possible face variations in the bag of faces and further balance the trade-off between performance and retrieval time.

In order to evaluate the performance of the proposed methods, we conduct extensive experiments on two real-world datasets. One of the datasets is constructed from TRECVid [7] videos during 2004 to 2006; another dataset is constructed from a Japanese news program "NHKnews7" during 2001 to 2011. These datasets contain faces in unconstrained envi-

B.-C. Chen (e-mail: sirius42@cmlab.csie.ntu.edu.tw) and Y.-Y. Chen (e-mail: yanying@cmlab.csie.ntu.edu.tw) are with the Department of Computer Science and Information Engineering, National Taiwan University. Y.-H. Kuo (e-mail: kuonini@cmlab.csie.ntu.edu.tw) and W. H. Hsu (e-mail: winston@csie.ntu.edu.tw) are with the Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan. T. D. Ngo (e-mail: ndthanh@nii.ac.jp), D.-D. Le (e-mail: ledduy@nii.ac.jp) and S. Satoh (e-mail: satoh@nii.ac.jp) are with National Institute of Informatics, Tokyo, Japan. Prof. Hsu is the contact person.

ronments [1] and are really challenging for content-based face retrieval. In the experiments we show that the proposed method can achieve significant performance gains over the prior state-of-the-art face recognition methods for face tracks or image sets while maintaining an highly scalable structure.

To sum up, our contributions include:

- We propose an novel coding method to encode the face track as a bag-of-faces sparse representation to solve face track retrieval problem in large-scale videos.
- We generalize the proposed coding method to enable multiple sparse representations for bag of faces, accommodate possible face variations, and balance the trade-off between performance and retrieval time.
- We conduct extensive experiments by the proposed methods on two face track datasets constructed from real-world videos and compare the results with state-of-the-art face retrieval methods for image sets. The datasets are publicly available [2] for future studies on face retrieval in videos.

II. Related work

Faces are always the subjects of interest for researchers because they are close to our daily life. Although studies on face recognition have shown promising on datasets in controlled environments, performance on real-world datasets is still unsatisfactory because face appearances have large variations in pose, expression, illumination, etc.

To overcome this issue, recently many studies focus on face recognition from sets of images. Instead of recognizing people using single image, they use a set of face images from the same person for recognition. In [8], X. Liu and T. Chen use adaptive Hidden Markov Models to model the faces extracted from the videos. In [9], Lee et al. use probabilistic appearance manifolds to model the faces. Some studies represent a set of images as a parametric distribution function such as Gaussian [10] or Gaussian Mixture Model [11] and use KL-Divergence to measure the distance between two sets. Some other studies use linear subspace [2], [12], [13] or mixture of linear subspace [14], [15] to represent a set of faces and use principal angles [16] to measure the distance between the two subspaces. In [4], Satoh proposes to use minimum distances between samples in two sets as the set distance; a similar idea is adopted by Cevikalp and Triggs [1], but instead of directly using samples in the set, they model the image set using an affine hull and find the closest points in the affine hull by solving a convex optimization problem. Hu et al. [17] further propose to find sparse approximated nearest point distance between points in affine hull to improve the performance. Although many effective methods are proposed to compute distance between two set of face images, they all ignore the scalability issues including (1) retrieval efficiency (by linear search versus by indexing), (2) the memory consumption (dense features versus sparse features), (3) similarity measurement (real values versus binary values), which should be considered to meet the scalability requirements of online large-scale retrieval system. Therefore, these methods can not be directly applied for face track retrieval in videos as the dataset grows.

Recently, some studies are trying to solve content-based face image retrieval problem. In [18], Wu et al. propose an identity-based quantization method for large-scale face image retrieval. Theodorakopoulos et al. [19] propose local sparse coding to represent a face by patch-based overcomplete dictionaries and to express pairwise similarities between faces. Chen et al. [20] propose to use sparse coding with identity constraints to improve the retrieval performance. Motivated by these methods, we propose to use bag-of-faces sparse representation to represent a face track extracted from the video. A face track is represented by a single sparse representation using the proposed method, and therefore efficient indexing method (i.e. inverted indexing) can be directly applied on large-scale dataset for real-time face track retrieval in large-scale videos.

III. System overview

We first use the face tracking method proposed in [21] to track faces in the videos, faces in the same track are grouped as a bag of faces. For each face in one bag, we apply facial landmark detection and extracted 149 dimension pixel-wise features at 13 different landmark locations to describe the faces as in [3]. Methods described in Section IV are then used to encode each bag of faces into one or more sparse representations. Inverted indexing is then built using non-zero entries in sparse representations as codewords for better performance and efficiency in retrieval [22], [23]. The system diagram is illustrated in Figure 2.

IV. Proposed method

For construction of face tracks, we take temporal information to extract faces shown in consecutive video frames. Note that, within a track, we do not consider their temporal orders because faces of a person in different tracks comprise different expressions and motion, which have no exact correspondences between their temporal orders. In the following subsections, we first describe how to find sparse representation of a single (face) image using sparse coding. Secondly, we describe how we generalize sparse coding framework to find sparse representation of a bag of faces. Finally, we describe how to improve retrieval performance by using multiple sparse representations for a bag of faces when the bag of faces contains large variations.

A. Sparse representation for single face image (SR)

Sparse representation has been proved very effective for face related work. Wright et al. [24] propose to use sparse representation for face recognition and achieve state-of-the-art performance. In [20], Chen et al. propose to use sparse representation of image patch as codewords for face image retrieval and demonstrated its effectiveness over prior common features in two open benchmarks. Here we show how to derive the sparse representation of a single face image for face image retrieval as shown in Figure 2(a). Let p be the number

[1] Unconstrained environments mean that the wild photos are taken in real life where the parameter settings of environment, e.g., lighting, angle, position, are unconstrained.

[2] http://satoh-lab.ex.nii.ac.jp/users/ndthanh/NIIFacetrackDatasets/

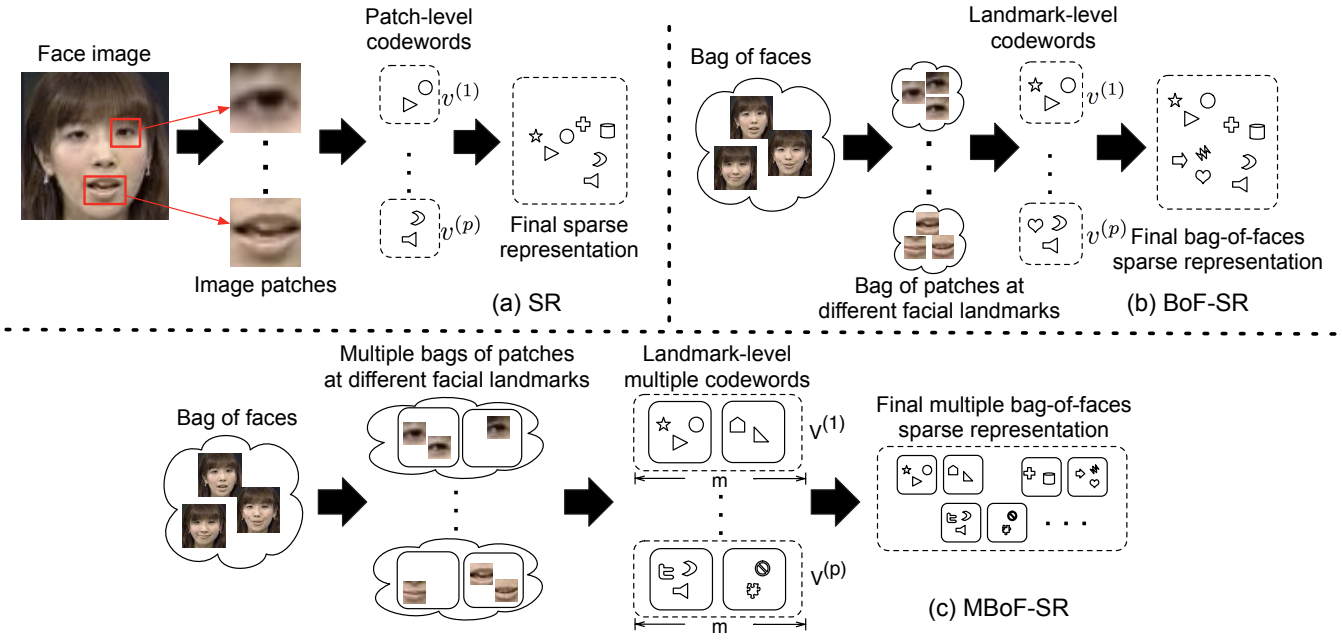

Fig. 2. (a) Using sparse coding for face retrieval with still image. Several patches are extracted from a face image at different facial landmarks (e.g., eyes corners, nose tips, mouth corners, etc.). For each patch, a sparse representation $v^{(i)}$ is found using Equation (1). All sparse representations are then concatenated together to form the final sparse representation to describe the face image. (b) The proposed bag-of-faces sparse representation method for face tracks. Patches extracted from the same facial landmark in bag of faces (from the same face track) are grouped together as a bag of patches and are used to find a sparse representation by Equation (2). Sparse representations at different locations are then concatenated together to form the final representation for the bag of faces. (c) Because the bag might contain faces with large variations, multiple sparse representations (indexed by m) are computed based on Algorithm 1 at each facial landmark. For instance, two sparse representations can be found to represent the bag of faces at the mouth location; in an automatic and approximate manner, one is used to represent faces in the bag with mouth closed and the other is with mouth opened. All sparse representations are aggregated together to represent the bag of faces. Equation (6) is then adopted to compute the distance between two bags (i.e., face tracks). Note that the sparse codewords can be indexed to facilitate large-scale face retrieval in video archives.

of landmark location in faces. Given a set of p dictionaries used to encode the p image patches and 149-dimensional pixel-wise features extracted from these patches, we find a sparse representation for each patch by solving the following optimization problem:

$$\underset{v^{(1)}\dots v^{(p)}}{\text{minimize}} \quad \sum_{i=1}^{p}(||x^{(i)} - D^{(i)}v^{(i)}||_2^2 + \lambda||v^{(i)}||_1), \qquad (1)$$

where p is the total number of patches in the face image, $x^{(i)}$ is the feature vector extracted from patch at location i (e.g., left eye and nose) of the image, $D^{(i)} \in R^{d\times k}$ is a dictionary contains k codewords with d dimensions and is used to encode the patch extracted from location i of the face. $v^{(1)}, v^{(2)}, \dots, v^{(p)}$ are the sparse representations of the image patches from location $1, 2, \dots, p$ respectively. Since the objective function is convex over $D^{(i)}$ while $v^{(i)}$ is fixed and vice versa. We solve the optimization problem by iteratively minimizing $D^{(i)}$ and $v^{(i)}$ by an efficient online algorithm [25]. Using sparse coding, a patch feature is encoded as a sparse linear combination of the column vectors of the dictionary. After the sparse representations are found, each non-zero entries of $v^{(i)}$ is considered as a codeword of the image for inverted indexing; note that the positive and negative value are consider as different codewords and the dimension of $v^{(i)}$ is k, therefore the size of the vocabulary (number of different codewords) is $2 \times p \times k$. The above problem is a set of unconstrained L1-regularized least square problem which can be solved efficiently using many different algorithms such as LARS [26].

B. Bag-of-faces sparse representation (BoF-SR)

For bags of faces, because number of faces is different in each bag, instead of finding a sparse representation for each patch, we propose to aggregate all the patches extracted from the same location and find a sparse representation for each bag-of-patches at certain location as shown in Figure 2(b). To find the sparse representation at each location, we solve the following optimization problem generalize from Equation (1):

$$\underset{v^{(1)}\dots v^{(p)}}{\text{minimize}} \quad \sum_{i=1}^{p}(\frac{1}{n}\sum_{j=1}^{n}||x_j^{(i)} - D^{(i)}v^{(i)}||_2^2 + \lambda||v^{(i)}||_1), \qquad (2)$$

where n is the number of faces in the bag, $x_j^{(i)}$ is the feature extracted from j_{th} face at location i. By solving the above optimization problem, $x_1^{(i)}, x_2^{(i)}, \dots, x_n^{(i)}$ are represented by a single sparse representation $v^{(i)}$ where $v^{(i)}$ minimize the average of reconstruction error for all patches at location i in the bag. The idea is to find a best sparse representation $v^{(i)}$ to encode all the patches at certain location in the bag of faces. Each $v^{(i)}$ in the above problem can be solved separately with an unconstrained L1-regularized sum of least square problem, which can be viewed as a larger L1-regularized least square

(a) (b)

Fig. 3. Two examples of bag of faces that are hard to represent by single sparse representation. (a) The bag of faces contains two facial expressions – looking at the camera and looking at the script. (b) The bag of faces contains some noises due to possible tracking errors. In these cases, using multiple bag-of-faces sparse representations (i.e., codebooks) can achieve better performance.

problem and can also be solved with LARS algorithm [26]. Note that when there is only one face in the bag, the above problem is reduced to Equation (1). The size of the vocabulary for a bag of faces is the same as the case in single image, and the size of database is reduced from millions of faces to tens of thousands of bags. Therefore, we can achieve very efficient online retrieval response.

C. Multiple sparse representations for bag of faces (MBoF-SR)

Using the above method, we can find a sparse representation of each bag of faces and achieve efficient retrieval speed, but sometimes a single sparse representation can not well characterize all the patches at a single location. Figure 3 shows two failure cases. Figure 3 (a) is a bag of faces extracted from a news video with a person in speech. There are two types of expressions in the bag of faces, one is when the person is looking at the camera, the other is when she is looking at the scripts. Figure 3 (b) is another bag of faces containing the same person; in the bag of faces, some of the faces are noisy due to face tracking errors. In these two cases, some patches extracted at the same facial landmark are quite different, therefore, we propose to use *multiple* sparse representations to represent the bag of faces where each sparse representation is used to represent a subset of the patches in the bag of patches at certain landmark location as shown in Figure 2 (c). We formulate this into the following optimization problem:

$$\begin{aligned} \underset{V^{(i)},S^{(i)},\forall i}{\text{minimize}} \quad & \sum_{i=1}^{p}\Big(\frac{1}{n}\sum_{j=1}^{n}||x_j^{(i)} - D^{(i)}V^{(i)}s_j^{(i)}||_2^2 \\ & + \lambda\sum_{k=1}^{m}||v_k^{(i)}||_1\Big) \\ \text{subject to} \quad & ||s_j^{(i)}||_0 = 1, ||s_j^{(i)}||_1 = 1, s_j^{(i)} \geq 0, \forall i, j, \end{aligned} \tag{3}$$

where $V^{(i)} = [v_1^{(i)}, v_2^{(i)}, \ldots, v_m^{(i)}]$ are m sparse representations for patches at location i, $S^{(i)} = [s_1^{(i)}, s_2^{(i)}, \ldots, s_n^{(i)}]$, and $s_j^{(i)} \in \{0,1\}^m$ is a zero-one vector indicating which column of $V^{(i)}$ is used to represent $x_j^{(i)}$. For instance, if $s_j^{(i)} = e_2$[3], then $V^{(i)}s_j^{(i)} = v_2^{(i)}$; therefore, $x_j^{(i)}$ is reconstructed by $D^{(i)}v_2^{(i)}$. The idea is to find multiple sparse representations and each of the representation can represent a subset of patches in the bag of patches that contains large variations. By minimizing the above objective function, we simultaneously find multiple sparse representations for bag-of-patches at each location ($V^{(i)}$) and decide the sparse representations are used to represent which patches ($S^{(i)}$).

The above problem is not convex because the feasible set (i.e. the set contains all the possible solution that satisfy the constraints in the problem) is not convex; therefore it is hard to find optimal solution of this problem. Here we propose an algorithm to find a suboptimal solution by iterative minimize $V^{(i)}$ and $S^{(i)}$.

When $S^{(i)}$ is fixed in the Equation (3), we can find each column of $V^{(i)}$ separately by solving the following unconstrained convex optimization problem:

$$\underset{v_j^{(i)}}{\text{minimize}} \quad \frac{1}{n}\sum_{k,\forall s_k^{(i)}=e_j}||x_k^{(i)} - D^{(i)}v_j^{(i)}||_2^2 + \lambda||v_j^{(i)}||_1, \tag{4}$$

when $V^{(i)}$ is fixed, we can find each $s_j^{(i)}$ by solving the following optimization problem:

$$\begin{aligned} \underset{s_j^{(i)}}{\text{minimize}} \quad & ||x_j^{(i)} - D^{(i)}V^{(i)}s_j^{(i)}||_2^2 \\ \text{subject to} \quad & ||s_j^{(i)}||_0 = 1, ||s_j^{(i)}||_1 = 1, s_j^{(i)} \geq 0. \end{aligned} \tag{5}$$

The size of feasible set in Equation (5) is only m; therefore, we can solve it by simply trying all possible value for $s_j^{(i)}$. The algorithm for solving Equation (3) is summarized in Algorithm 1. In each iteration, we alternatively divide the bag of patches at each location into different subset using $S^{(i)}$ and find the suitable sparse representation for each subset of patches. The algorithm will converge because in each iteration the objective function in Equation (3) will decrease and there is only a finite set of possible $S^{(i)}$. Note that although the algorithm will converge, it does not guarantee to find the optimal solution, and the result depends on the initial value of $S^{(i)}$, but we find that in practice we can usually find a good set of sparse representations for the bag of faces and will converge in several iterations.

After the above procedure, each bag is represented by p sets of sparse representations, $B_1 = \{V_1^{(1)}, \ldots V_1^{(p)}\}$, $B_2 = \{V_2^{(1)}, \ldots V_2^{(p)}\}$, the similarity between two bags is then defined as follow:

$$S(B_1, B_2) = \sum_{i=1}^{p}\max_{j,k} c(v_{1,j}^{(i)}, v_{2,k}^{(i)}), \tag{6}$$

where $c(a,b)$ indicates the number of overlapping codewords between two sparse vectors,

$$c(a,b) = ||\max((a \circ b), \mathbf{0})||_{\mathbf{0}}, \tag{7}$$

"$\circ$" denote the element-wise multiplication between two vectors. Note that using Equation (7), only coding value with the

[3]Here e_i is a m dimensional vector with all zeros except i_{th} dimension is one as defined in most linear algebra literature.

same sign will be considered as the same codewords. That is, we consider coding value with different signs as different codewords. By considering the sign of coding values, we effectively get sparse representation with $2 \times k$ dimensions. It can be viewed as sparse coding using a larger dictionary $[-D\ D]$ with $2 \times k$ entries. Equation (6) computes the sum of maximum number of overlapping codewords at each location between two bags of faces.

To efficiently compute the similarity measure in Equation (6), we use a modified version of inverted index. For each entry in inverted list, we maintain a Bag-ID that denotes which bag this codeword belongs to, and a Representation-ID, ranging from 1 to m, denotes which sparse representation of the bag this codeword comes from. For each sparse representation in query face track, we retrieve the index and compute the number of overlapping codewords between query and every sparse representation in the index and will derive m different scores for each Bag-ID. We keep the best score among these m scores. After m runs with different sparse representations in query face track, we can find the maximum number of overlapping codewords between query sparse representations and sparse representations in the index. Since the number of sparse representation is m times more than the case with single sparse representation, the average length of posting lists in inverted index is m times longer; therefore, it takes m^2 time to retrieve the index and compute the score.

Algorithm 1 Algorithm for finding sets of sparse representations

Input: A set of dictionaries $D^{(1)}, \ldots, D^{(p)} \in R^{d \times k}$; features extracted from the bag of faces at each location $X^{(1)}, \ldots, X^{(p)} \in R^{d \times n}$;$n$ (the size of the input bag of faces); m (the number of output sparse representations)

Output: A set of sparse representations for each location $V^{(1)}, \ldots, V^{(p)} \in R^{k \times m}$;

```
for i = 1 to p do
    Randomly choose S^(i) that satisfy the constraint in Equation (3)
    repeat
        for j = 1 to m do
            Solving Equation (4) using LARS algorithm
        end for
        for j = 1 to n do
            Solving Equation (5) by trying all elements in feasible set
        end for
    until converge
end for
```

V. Experiments

A. Dataset

We use two different datasets to evaluate our system. The first one is extracted from TRECVid [7] news videos during 2004 to 2006. Around 20 millions faces are detected from the videos; the detected faces are then tracked and grouped as around 157K bags of faces. As reported in Table I, 1,497 bags of faces with 405K faces from 41 well known people are annotated for evaluation. The second one is extracted from a news program broadcast in Japan "NHK news7" during 2001 to 2011. 5,567 bags of faces with 1.25 million faces from 111 people are annotated for evaluation. To our best knowledge, this dataset is one of the largest datasets available for face track retrieval task and is really challenging because it contains not only variations from illumination, pose, expression variation but also biological variations between faces of the same person due to long time period. Throughout the experiments for each dataset, in a leave-one-out manner, each bag of faces is alternatively used as query while remaining bags are used as database for computing the average precision. Mean average precision (MAP), which is a common measurement adopted in many literatures for retrieval task, is then computed for all queries.

TABLE I
THE STATISTICS OF EXPERIMENTAL DATASETS – TRECVID AND NHK NEWS VIDEOS; THE DETAILS ARE EXPLAINED IN SECTION V-A.

Datasets	Annotated Tracks	Faces	Identities
TRECVid	1,497	405K	41
NHKnews7	5,567	1.25M	111

B. Compared algorithms

We compare our methods to several state-of-the-art methods in the experiments including:

- SR: patch-level sparse representation from a single image as shown in Figure 2 (a) [20]. We simply pick the first face in the bag to represent the bag of faces. This baseline is used to illustrate the effectiveness of the bag-of-faces representation.
- MSM: mutual subspace method proposed in [2]. We first use PCA to find subspace bases and use the average of top ten canonical correlations for computing distance.
- Min-Min: minimum distance between samples from two sets as proposed in [4]. We use one minus cosine similarity as our distance measure.
- AHISD: affine hull image set distance proposed in [1].
- CHISD: convex hull image set distance proposed in [1]. For both AHISD and CHISD, we use the linear version and retain 98% energy by PCA. For CHISD, we set $C = 100$ for SVM training as in [1].
- BoF-SR: bag-of-faces sparse representation proposed in this paper.
- MBoF-SR: multiple bag-of-faces sparse representation proposed in this paper.

Note that we use the same pixel-wise feature extracted from 13 different facial landmarks for all the above methods. For SR, BoF-SR and MBoF-SR, we use random samples from NHKnews7 dataset as our dictionary entries. The advantage for using random sampling is the time efficiency to construct dictionaries of large size, which is a major superiority in representing large-scale visual data. In terms of representativeness, Coates and Ng [27] found that using randomly sampled image patches as dictionary can achieve similar performance as that

by using learned dictionary ($< 2.7\%$ relative improvement in their experiments) if the sampled patches provide a set of over-complete basis that can represent input data. Therefore, in the experiments, we adopt random sampling to efficiently obtain a large dictionary to meet the needs of representing large-scale video data.

For MSM, Min-Min, AHISD and CHISD, we use linear search to derive ranking results, since these methods cannot well work with current indexing frameworks; for SR, BoF-SR and MBoF-SR, we use inverted indexing. For AHISD and CHISD, we use the MATLAB implementation provided by the author of [1], other methods (MSM, SR, BoF-SR, MBoF-SR) are carefully implemented in MATLAB; inverted indexing is implemented with C++. All the experiments operate on a 2.4 GHz Intel Xeon server.

C. Evaluation of the proposed methods

Table II shows the performance of the proposed methods compared to other state-of-the-art methods. In BoF-SR and MBoF-SR, we use $\lambda = 0.125$ and $k = 1000$. In MBoF-SR, we use eight sparse representations to represent the bag of faces ($m = 8$); discussions on the parameters will be shown in the following sections. SR performs much worse than other methods because it only uses single face image. Therefore, we can see that using bag of faces can really help the performance since it exploits more information. Among all other methods, Min-Min shows salient performance but it takes a long time for retrieval and, therefore, are not applicable for large-scale data. Note that AHISD performs worse than other baseline methods, it is because when size of the bags is big, affine hull representation might be too strong and every affine hull in the dataset is really close to each other in the feature space.

Using BoF-SR we can achieve 8% absolute improvement compared to other state-of-the-art methods on Trecvid dataset and 8.6% improvement on NHKnews7 dataset while having the fastest online retrieval time (0.01s on Trecvid dataset). MBoF-SR can further improve the performance by 2.5% on Trecvid dataset and 5.2% on NHKnews7 dataset and still have a reasonable online retrieval time. Table II also shows the top rank precision (P@10) of all the methods. The proposed methods can achieve not only the best MAP compared to other methods but also best P@10 on both datasets. Note that the time shows in the Table II only contains the online retrieval time and does not include the time for face tracking, feature extraction and computing representation for bag of faces because the time is independent with the size of the datasets.

For MSM, Min-Min, AHSID and CHISD, we need to use linear search to derive the ranking results; therefore, the retrieval time is linear to the dataset size. On the other hand, SR, BoF-SR, MBoF-SR can achieve sub-linear retrieval time by using inverted indexing. For AHISD and CHISD, the retrieval time on NHKnews7 is too long. For a single query, it takes more than 1,000 seconds to finish. To evaluate all 5,567 queries in the dataset, it will take more than two months to finish; therefore, we do not show the performance here.

[4]Because the features used in [28] (Local Binary Pattern) are different from those in this work (pixel-wise features), the results of MSM are slightly different.

D. Impact of the parameters in BoF-SR

Figure 4 shows the performance of BoF-SR using different parameters on two datasets. We run experiments with λ from 0.125 to 4, k from 200 to 1000 on both datasets. We find that when k is too small (green line with cross in Figure 4) the performance is worse because the dictionary does not have enough discriminative capability to represent the bag of faces. When dictionary is large enough, the performance is similar regardless the size of the dictionary. The performance is better when λ is small, because more (denser) codewords are used to represent the bag of faces. However, the performance tends to saturate when λ is too small because the sparsity of the representations tends to stay the same. Note that there is a trade-off between performance and retrieval time, when λ is large, the number of codewords will drop because the sparsity of the representations increase, therefore, the retrieval time is faster. Throughout the following experiments, we set $\lambda = 0.125$ and $k = 1000$ for both BoF-SR and MBoF-SR on both datasets.

E. Number of sparse representations in MBoF-SR

In Figure 5, we show the performance and retrieval time on MBoF-SR using different m. When m increases, the performance on both datasets increases; it evidences the effectiveness of the proposed MBoF-SR. When there are more sparse representations used to represent the bag of faces, it can be described better. The retrieval time required by MBoF-SR is m^2 times compared to BoF-SR, so when m increases, retrieval time also increases. Nevertheless, when m is small, the retrieval time increases slowly while the performance gains more significantly; therefore, we can choose a small m to achieve better performance with a reasonable retrieval time.

F. Size of the bag

We also conduct experiments by varying bag sizes (i.e., numbers of faces per bag). Table III shows the MAP performance on Trecvid dataset with different bag sizes. We take first n faces from the bag of faces to compute the result. If the total number of faces is smaller than n, all faces are used for the experiments. We gain more performance gains as increasing the bag size since more redundancies in faces can be exploited. However, the proposed methods consistently have the best performance for all different sizes. Performance on AHISD and CHISD drops when using all the images in the bags. This is probably because when size of the bag is huge, affine hull or convex hull representation is too strong and most of the bags become really close to each other in the feature space and thus lacks discriminative capability. Also note that the retrieval time required by the proposed methods (BoF-SR, MBoF-SR) does not change much while it increases dramatically when using Min-Min, AHISD and CHISD. The scalability and effectiveness are both ensured in the proposed methods. The example retrieval results by the proposed MBoF-SR are shown

TABLE II

COMPARISONS WITH DIFFERENT METHODS ON THE TWO VIDEO DATASETS. USING SINGLE FACE OBTAIN WORSE PERFORMANCE COMPARED WITH THOSE USING A BAG OF FACES BECAUSE THE LATTER CONTAINS MORE VISUAL CUES. THE PROPOSED METHODS (BOF-SR, MBOF-SR) ACHIEVE BETTER PERFORMANCE IN TERMS OF MAP AND P@10 COMPARED WITH ALL OTHER PRIOR METHODS FOR MEASURING SIMILARITIES BETWEEN BAGS OF FACES. THE PROPOSED METHODS ALSO CONSUME MUCH LESS TIME FOR RETRIEVAL. NOTE THAT ALL OTHER METHODS (MSM, MIN-MIN, AHISD AND CHISD) REQUIRE LINEAR SEARCH TO DERIVE RANKING RESULTS AND THEREFORE THE (AVERAGE) RETRIEVAL TIME (PER QUERY) GROWS LINEARLY WITH THE SIZE OF DATASET; METHODS USING SPARSE REPRESENTATIONS (SR, BOF-SR AND MBOF-SR) BENEFIT FROM INVERTED INDEXING AND CAN ACHIEVE SIGNIFICANT RETRIEVAL EFFICIENCY.

Dataset/Methods	Trecvid			NHKnews7		
	MAP	P@10	Time (s)	MAP	P@10	Time (s)
SR [20]	49.8%	74.5%	**0.01**	22.5%	63.5%	**0.03**
MSM^4[2]	72.3%	86.6%	0.83	56.3%	87.2%	2.84
Min-Min [4]	75.9%	88.0%	128	59.8%	91.3%	342
AHISD [1]	55.6%	73.9%	347	–	–	est. 1041
CHISD [1]	67.6%	83.8%	639	–	–	est. 1917
BoF-SR	83.9%	91.3%	**0.01**	68.4%	94.7%	**0.03**
MBoF-SR ($m = 8$)	**86.4%**	**92.4%**	0.59	**73.6%**	**95.5%**	1.59

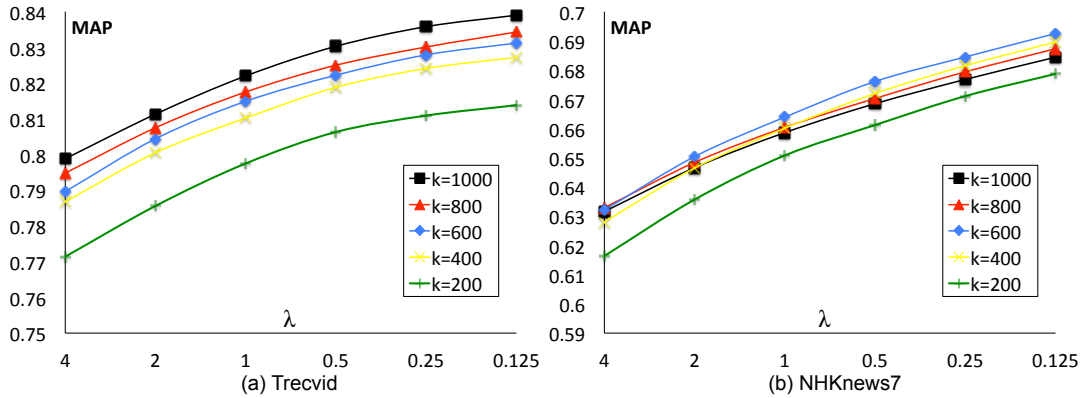

Fig. 4. The performance of the proposed BoF-SR with different parameters on the two datasets. When the size of the dictionary is too small (e.g., $k = 200$), the performance on the both datasets becomes worse due to limited discriminative capability. When λ (in Equation 4) decreases, the accuracy (MAP) improves on the both datasets because more codewords are used to represent the bag of faces. There is a trade-off between performance and retrieval time; more codewords will affect the retrieval efficiency. Note that, for these parameterizing setups, the proposed methods still outperform the prior state-of-the-art methods.

TABLE III

THE COMPARISONS (ACCURACY AND EFFICIENCY) WITH DIFFERENT PRIOR METHODS WITH VARYING BAG SIZE (NUMBER OF FACES) ON TRECVID DATASET. NUMBERS IN THE PARENTHESES ARE THE AVERAGE RETRIEVAL TIME (IN SECOND AND PER QUERY). WHEN THE BAG SIZE IS SMALL, THE PERFORMANCE OF ALL METHODS DROPS BECAUSE FEW FACE CUES ARE EXPLOITED. HOWEVER, THE PROPOSED METHODS CONSISTENTLY ACHIEVE THE BEST PERFORMANCE WITH ALL DIFFERENT BAG SIZES. NOTE THAT THE RETRIEVAL TIME FOR MIN-MIN, AHISD AND CHISD INCREASES WITH THE SIZE OF THE BAGS AND IS NOT FEASIBLE FOR LARGE-SCALE VIDEO ARCHIVES. ON THE OTHER HAND, THE RETRIEVAL TIME FOR THE PROPOSED METHODS DO NOT CHANGE MUCH WITH THE BAG SIZE.

Bag Size/Methods	MSM [2]	Min-Min [4]	AHISD [1]	CHISD [1]	BoF-SR	MBoF-SR
25	66.0% (0.86s)	69.6% (11.1s)	68.5% (12.3s)	68.1% (13.5s)	76.9% (**0.01s**)	**80.5%** (0.28s)
50	69.7% (0.87s)	72.9% (21.3s)	70.3% (35.5s)	70.4% (33.4s)	79.9% (**0.01s**)	**83.2%** (0.29s)
100	71.4% (0.98s)	75.3% (41.4s)	70.8% (76.5s)	71.7% (64.9s)	82.0% (**0.01s**)	**85.0%** (0.30s)
All	72.3% (0.83s)	75.9% (128s)	55.6% (347s)	67.6% (639s)	83.9% (**0.01s**)	**86.4%** (0.59s)

in Figure 7. A query bag ((a) or (b)) usually carries diverse face appearances of the same person with apparent visual variations in poses, facial expressions, etc. The proposed MBoF-SR can comprehensively depict these intra-class variations, e.g., varied poses in query (a) and different facial expressions in query (b), by multiple sparse representations and further improve the retrieval accuracy compared to retrieval by single representation (cf. Table II).

G. Time of Encoding Face Track

TABLE IV

THE TIME FOR ENCODING A FACE TRACK WITH A VARYING NUMBER OF MULTIPLE SPARSE REPRESENTATIONS. M INDICATES THE NUMBER OF SPARSE REPRESENTATIONS AND TIME IS MEASURED BY SECONDS.

m	1 (BoF-SC)	2	3	4	5	6	7	8
Time	1.1	5.8	14.1	17.5	24.4	28.4	35.3	36.8

Table IV shows the average encoding time required by MBoF-SC with different m (the number of sparse representa-

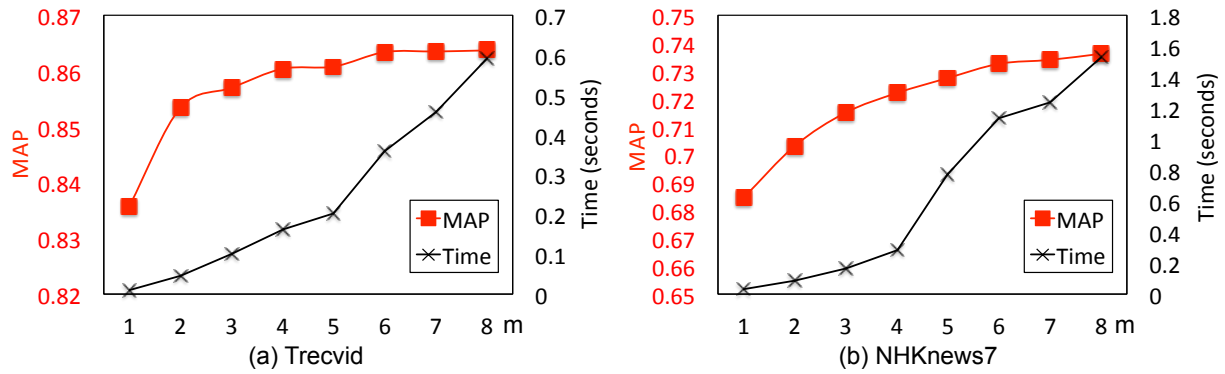

Fig. 5. The performance and retrieval time of the proposed MBoF-SR with different number of sparse representation models (m) for the bag of faces on the two datasets. Using MBoF-SR, we can achieve 2.5% and 5.2% absolute improvements in MAP respectively. When m is large, more sparse representations are considered and can better describe the bag of faces, however, with the cost of more retrieval time. When m is small (e.g., 2), the retrieval time is small but still brings significant improvement. m equals 2 or 3 might be a reasonable setup for balancing retrieval accuracy and efficiency for large-scale video archives.

TABLE V
THE ENCODING TIME WITH DIFFERENT BAG SIZES. THE ENCODING TIME INCREASES BECAUSE MORE ITERATIONS ARE REQUIRED FOR MEASURING THE CODEWORD RESPONSES; HOWEVER, THIS IS STILL ACCEPTABLE IN THE APPLICATION SCENARIO BECAUSE ENCODING CAN BE PROCESSED IN THE OFF-LINE STAGE.

bag size	25	50	100	full
Time	13.6	18.9	25.1	36.8

tions), and Table V shows the encoding time with different bag sizes as $m = 8$ and sparse representations are used. With either m or the bag size increases, the encoding time increases because more iterations are required for measuring the codeword responses. This is acceptable in the application scenario because encoding can be processed in the off-line stage. Note that $m = 1$ yields the results of BoF-SC.

H. Effect of dictionary construction

Figure 6 shows the comparisons between retrieval performance on Trecvid dataset using randomly sampled dictionary and learned dictionary with different dictionary size k by BoF-SR. For randomly sampled dictionary, we sample k face features from NHKnews7 dataset as dictionary entries. For learned dictionary, we sample $10,000$ face features from NHKnews7 dataset and use online dictionary learning algorithm in [25] to train our dictionary. As shown in Figure 6, when the dictionary size is small (e.g., $k = 200$), using the learned dictionary achieves better performance; this is because the learned dictionary is optimized to approximate the data. However, when dictionary size becomes larger, using randomly sampled dictionary can achieve similar performance as the learned one. This suggests that when dictionary size is large enough, randomly sampled dictionary can sufficiently represent the input data. The similar results can be found in [27].

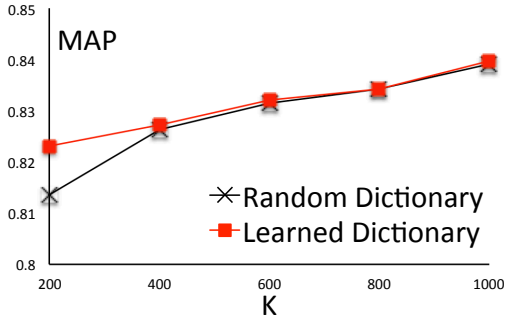

Fig. 6. Retrieval performance of Trecvid dataset using different dictionaries. When dictionary size is small (e.g., $k = 200$), using learned dictionary achieves better performance. However, when dictionary becomes larger, the randomly sampled dictionary can achieve similar performance as the learned one.

VI. CONCLUSIONS

To solve scalable face retrieval problem in large-scale video archives, we propose a effective and efficient method – bag-of-faces sparse representation to encode the bag of faces into discrete codewords. To further improve the performance, we generalize the bag-of-faces sparse representation to find multiple sparse representations in an unsupervised and automatic manner. Using the proposed methods, multiple bag-of-faces sparse representations and faces are encoded and simultaneously obtained by solving an optimization problem. Extensive experiments on two real-world datasets show that the proposed methods can not only achieve significant performance over the prior state-of-the-art methods but also require much less retrieval time.

REFERENCES

[1] H. Cevikalp and B. Triggs, "Face recognition based on image sets," *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.
[2] O. Yamaguchi, K. Fukui, and K.-I. Maeda, "face recognition using temporal image sequence," *International Symposium of Robotics Research*, 1998.

Fig. 7. Example retrieval results from Trecvid dataset (a) and from NHKnews7 dataset (b) using the proposed method MBoF-SR. Query bags of faces carry more diverse face appearances of the same identity because of the visual variations in pose, facial expression, lighting, etc. MBoF-SR can comprehensively depict these intra-class variations, e.g., varied poses in query (a) and different facial expressions in query (b), by multiple sparse representations and further improve the retrieval accuracy compared to retrieval by single representation (cf. Table II).

[3] M. Everingham, J. Sivic, and A. Zisserman, ""hello! my name is ... buffy" automatic naming of characters in tv video," *British Machine Vision Conference*, 2006.

[4] S. Satoh, "Comparative evaluation of face sequence matching for content-based video access," *IEEE International Conference on Automatic Face and Gesture Recognition*, 2000.

[5] T. N. Nguyen, T. D. Ngo, D.-D. Le, S. Satoh, B. H. Le, and D. A. Duong, "An efficient method for face retrieval from large video datasets," *International Conference on Image and Video Retrieval*, 2010.

[6] J. Sivic, M. Everingham, and A. Zisserman, "Person spotting: Video shot retrieval for face sets," *International Conference on Image and Video Retrieval*, 2005.

[7] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and trecvid," *ACM International Conference on Multimedia Information Retrieval*, 2006.

[8] X. Liu and T. Chen, "Video-based face recognition using adaptive hidden markov models," *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[9] K.-C. Lee, J. Ho, M.-H. Yang, and D. Kriegman, "Video-based face recognition using probabilistic appearance manifolds," *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[10] G. Shakhnarovich, J. W. Fisher, and T. Darrell, "Face recognition from long-term observations," *European Conference on Computer Vision*, 2002.

[11] O. Arandjelović, G. Shakhnarovich, J. Fisher, R. Cipolla, and T. Darrell, "Face recognition with image sets using manifold density divergence," *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[12] K. Fukui and O. Yamaguchi, "face recognition using multi-viewpoint patterns for robot vision," *International Symposium of Robotics Research*, 2003.

[13] T. Kim, J. Kittler, and R. Cipolla, "Discriminative learning and recognition of image set classes using canonical correlations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007.

[14] Ruiping Wang, S. Shan, X. Chen, and W. Gao, "manifold-manifold distance with application to face recognition based on image set," *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[15] R. Wang and X. Chen, "Manifold discriminant analysis," *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[16] H. Hotelling, "Relations between two sets of variates," *Biometrika*, 1936.

[17] Y. Hu, A. Mian, and R. Owens, "Sparse approximated nearest points for image set classification," *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[18] Z. Wu, Q. Ke, J. Sun, and H.-Y. Shum, "Scalable face image retrieval with identity-based quantization and multi-reference re-ranking," *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[19] I. Theodorakopoulos, I. Rigas, G. Economou, and S. Fotopoulos, "Face recognition via local sparse coding," *International Conference on Computer Vision*, 2011.

[20] B.-C. Chen, Y.-H. Kuo, Y.-Y. Chen, K.-Y. Chu, and W. Hsu, "Semi-supervised face image retrieval using sparse coding with identity constraint," *ACM Multimedia*, 2011.

[21] T. D. Ngo, D.-D. Le, S. Satoh, and D. A. Duong, "Robust face tracking finding in video using tracked points," *Proc. Intl. Conf. on Signal-Image Technology and Internet-Based Systems*, 2008.

[22] Hervé Jégou and Matthijs Douze and Cordelia Schmid, "Packing bag-of-features," *International Conference on Computer Vision*, 2009.

[23] Jun Yang and Yu-Gang Jiang and Alexander G. Hauptmann and Chong-Wah Ngo, "Evaluating Bag-of-Visual-Words Representations in Scene Classification," *International Workshop on Multimedia Information Retrieval*, 2007.

[24] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.

[25] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," *International Conference on Machine Learning*, 2009.

[26] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Annals of statistics*, 2004.

[27] Adam Coates and Andrew Ng, "The importance of encoding versus training with sparse coding and vector quantization," *Proceedings of the 28th International Conference on Machine Learning*, 2011.

[28] H. T. Vu, T. D. Ngo, T. N. Nguyen, D.-D. Le, S. Satoh, B. H. Le, and D. A. Duong, "Fast face sequence matching in large-scale video databases," *IEEE International Conference on Image Processing*, 2011.