2016 23rd International Conference on Pattern Recognition (ICPR)
Cancún Center, Cancún, México, December 4-8, 2016

Efficient Video Face Recognition by using Fisher Vector Encoding of Binary Features

Yoanna Martínez-Díaz*, Leonardo Chang*, Noslen Hernández†, Heydi Méndez-Vázquez* and L. Enrique Sucar‡
*Advanced Technologies Application Center
7a #21406 b/ 214 and 216, P.C. 12200, Playa, Havana, Cuba
Email: {ymartinez,lchang,hmendez}@cenatav.co.cu
†Pontifical Catholic University of Rio de Janeiro
Rua Marquês de São Vicente 225, Gávea, Rio de Janeiro, Brasil
Email: noslenh@gmail.com
‡Instituto Nacional de Astrofísica, Óptica y Electrónica
Luis Enrique Erro No.1, Tonantzintla, Puebla, México. CP 72840
Email: esucar@inaoep.mx

***Abstract*—One of the main problems of recognizing faces in videos is to achieve accurate algorithms which can be used in real-time applications. Recently, Fisher Vector representation of local descriptors (e.g., SIFT) has gained widespread popularity, achieving good recognition rates. In this work, we propose to use Fisher Vector encoding of binary features for video face recognition, in order to speed up the computation time of the representation. The experimental evaluation was conducted on the challenging *YouTube Faces* database, showing that the proposed method is very efficient, and has an accuracy comparable with state-of-the-art methods.**

I. Introduction

Video face recognition has become one of the most active research topics in the last years, mainly due to the great amount of videos available in online social media, surveillance systems and mobile devices [1], [2]. Compared with a single image, a video contains multiple frames of the same person, which can allow us to obtain face recognition systems more robust to pose variations and illumination changes. Nevertheless, it imposes new challenges, such as low resolution and motion blur. On the other hand, of particular importance is the development of efficient methods for handling videos in real-time applications, especially when both computing and memory resources are limited.

Several methods have been proposed for face recognition in videos [1], [3]–[6]. Pioneering works applied existing image based face recognition techniques to all (or selected) video frames and, then fused the matching results for the different frames when comparing two video faces [6]. However, this is not an efficient solution when videos have a large number of frames, and these methods are not able to represent all the variability that appears for one person in a video sequence. Other previous works represent face videos using super-resolution, sparse representations, 3D or manifold models [1], but they are in general computationally expensive. Spatio-temporal approaches emerged then as a solution for video face recognition, combining spatial and temporal cues, and increasing the efficiency of tracking plus recognition. These methods achieve good performance [5]. However, the recent spread of deep learning methods has taken video face recognition (as well as other applications) to a new level, significantly improving the state of the art [3], [4]. The basic idea of these methods is the use of Convolutional Neural Networks (CNN) in order to learn discriminative face features. Although there are differences among the existing CNN architectures, they have in common the requirement of a massive amount of labeled training data (millions of images) to generalize well, demanding thus a lot of training time.

There is another group of new "not deep" approaches that have also shown very good performance on video face recognition [7]–[10]. They are mainly based on two recently proposed methods: the probabilistic elastic part (PEP) representation [11] and the Fisher Vector (FV) representation [12]. In the original PEP model [11] each face image is partitioned into overlapping patches at multiple scales and local features (LBP [13] or SIFT [14]) are extracted from them. These local features are then augmented with its location in the face image, and a set of face part models are unsupervisedly learned. The Eigen-PEP [9] was later proposed to represent better the videos and recently, the Hierarchical-PEP [8] was introduced to increase its discriminative power in front of pose variations. On the other hand, the popular FV representation was first used for face recognition in [15] and it was later extended to be used in videos with the Video Fisher Vector Faces (VF^2) descriptor [10]. The FV representation is obtained by densely computing SIFT features (in an image or video) at different scales, which are aggregated into a high-dimensional vector by fitting and encoding the derivatives of the log-likelihood of a Gaussian Mixture Model (GMM).

As we can see, both PEP and FV representations have a common step, which consists of computing a set of local features based on SIFT descriptor. Although SIFT descriptor has proven to be highly discriminant, it requires an intensive computational effort, especially for real-time systems, such as the case of video face recognition. Recently, it has emerged a trend to increase the development of simpler descriptors with lower computation demands, focusing not only on perfor-

978-1-5090-4846-5/16/$31.00 ©2016 IEEE

mance but also on speed. Among the recent proposals, binary descriptors appear as a suitable option for real applications, i.e. BRIEF [16], ORB [17], BRISK [18] and FREAK [19] descriptors.

In this paper, we propose a method, namely BinVF^2 (Binary Video Fisher Vector Faces) for video face recognition, based on FV representation of binary features (instead of e.g., SIFT). The main motivation is to speed up the computation time of the representation and gain in efficiency, which is highly demanding for the task of video face recognition. Specifically, we choose Binary Robust Independent Elementary Features (BRIEF), which is two orders of magnitude faster than SIFT, and it has a simple construction and compact storage in memory. Since our features are binary, a generative model based on GMM is not appropriate. Instead, we use the derived Fisher Vector based on a multivariate Bernoulli Mixture Model (BMM) proposed by Uchida and Sakazawa [20]. In addition, we use the diagonal metric learning technique [15] in order to increase the discriminative power of the representation. The performance of the proposed method was evaluated, delivering a considerable speed-up, while maintaining a good trade-off between efficiency and accuracy with respect to state-of-the-art methods.

The remainder of this paper is organized as follows. Section 2 describes in detail the proposed representation for video face recognition. Section 3 provides a set of experiments and results on the public available *YouTube Faces* database [6]. Finally, conclusions and future work are given in Section 4.

II. The BinVF^2 Video Face Representation

As we mentioned, our video face representation is based on the Fisher Vector representation [12]. To build it for a video, local features are first densely extracted from all the video frames. In our case, we use the binary descriptor BRIEF to minimize the computational complexity of the representation. These features are modeled by using a mixture of a multivariate Bernoulli distributions [21]. In order to efficiently compute the posterior probabilities, we propose an index-based representation of the extracted descriptors. Entire information across multiple frames is merged by using a spatio-temporal pooling, similar to [10]. Thus, only a single Fisher Vector is computed for all local BRIEF descriptors extracted from the whole face video. A method overview of BinVF^2 is shown in Figure 1.

A. BRIEF descriptor

Although local features like SIFT and SURF have gain popularity, recently, binary descriptors have become a highly attractive solution for real-time applications. They are very fast to compute (two orders of magnitude faster than SIFT) and reduce memory consumption, resulting in more compact vectors.

Several binary descriptors such as BRIEF [16], BRISK [18] and FREAK [19] have been proposed in the literature. In this paper we choose BRIEF as local descriptor, due to its simple construction and compact storage in memory. Moreover, as

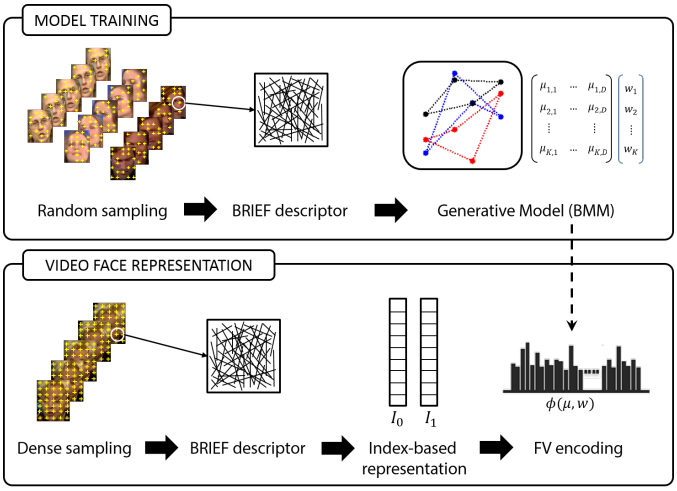

Fig. 1. Overview of the proposed method for video face representation.

shown in [22], BRIEF provides a very efficient technique for time-constrained applications with good matching accuracy, comparable with SIFT and SURF descriptors. Nevertheless, these advantages have not been exploited deeply on the case of video-based face recognition scenarios.

BRIEF represents an image patch as a binary string by using a set of binary tests. More specifically, let us consider a smoothed image patch I of size $M \times M$ then, a binary test τ is defined as:

$$\tau(I; x, y) := \begin{cases} 1 & \text{if } I(x) < I(y) \\ 0 & \text{otherwise,} \end{cases}$$

where x and y are locations in the patch I, and $I(x)$ denotes the pixel intensity of I at x. The locations of the pixels are pre-selected randomly according to a Gaussian distribution around the patch center. Finally, using a set of η binary tests, we obtain a bitstring of length η:

$$f_\eta(I) = \sum_{1 \le i \le \eta} 2^{i-1} \tau(I; x_i, y_i). \quad (1)$$

In this paper we use $\eta = 256$, since it has been shown that this value brings a good compromises between speed, storage efficiency, and recognition rate [16], [23]. The descriptors are extracted densely at multiple scales which represents an important advantage for the case of videos, because it is not necessary to run a keypoint detector on each frame which can be computationally expensive.

B. Fisher Vector encoding

Let $X = \{x_1, x_2, ..., x_T\}, x \in \mathcal{X}$, be a set of T ($T \in \mathbb{N}$) local D-dimensional descriptors. Let $p(x|\theta)$ be a probability density function which models the generation process of elements in $\mathcal{X}$. The FV of X, Φ^X, is defined as the normalized gradient $\Phi^X = L_\theta \nabla_\theta \log p(X|\theta)$, where L_θ is the matrix resulting from the following decomposition of the inverse of the Fisher Information Matrix $F_\theta^{-1} = L_\theta^T L_\theta$ for $p(x|\theta)$.

Due to the fact that our features are binary, i.e., $x_t \in \{0,1\}^D$, an appropriate parametric form for $p(x|\theta)$ is a Mixture of Bernoulli. In this way,

$$p(x|\theta) = \sum_{k=1}^{K} w_k p_k(x|\theta_k), \quad (2)$$

where K represents the number of components in the mixture and $p_k(x|\theta_k) = \prod_{d=1}^{D} \mu_{kd}^{x^d}(1-\mu_{kd})^{(1-x^d)}$ is a Multivariate Bernoulli distribution with parameter $\theta_k = (\mu_{kd}), d = 1, ..., D$.

The vector of parameters of the mixture is $\Theta = (w_k, \mu_{kd}), k = 1, ..., K, d = 1, ..., D$ which have dimension $K(D+1)$.

A closed-form approximation of the Fisher Vector for this particular setting was derived by Uchida and Sakazawa [20] in which,

$$\phi^X_{\mu_{kd}} = \left(\frac{1}{T} \sum_{t=1}^{T} \gamma_k(x_t) \frac{(-1)^{1-x_t^d}}{\mu_{kd}^{x_t^d}(1-\mu_{kd})^{\left(1-x_t^d\right)}} \right) F_{kd}^{-1/2}, \quad (3)$$

where

$$\gamma_k(x_t) = \frac{w_k p_k(x_t|\theta_k)}{\sum_{k=1}^{K} w_k p_k(x_t|\theta_k)}, \quad (4)$$

is the posterior (occupancy) probability of the sample x_t given the k-th component and

$$F_{kd} = T w_k \left(\frac{\sum_{k=1}^{K} w_k \mu_{kd}}{\mu_{kd}^2} + \frac{\sum_{k=1}^{K} w_k (1-\mu_{kd})}{(1-\mu_{kd})^2} \right). \quad (5)$$

Only the gradients with respect to μ_{kd} are taken into account, so the FV, $\Phi^X = (\phi^X_{\mu_{kd}}), k = 1, ..., K, d = 1, ..., D$ will have dimension KD. Note that compared with the FV based on GMM densities, the dimension is reduced by a half.

Finally, the FV is further improved by applying first power normalization and then renormalizing it by l_2 norm.

Attempting to improve the discriminative power by learning a good distance metric so that the distance between positive face pairs is reduced and that of negative pairs is enlarged as much as possible, a diagonal metric learning on the FV representation is used. This metric is carried out using a conventional linear SVM formulation [15].

C. Efficient posterior probabilities computation

The most computationally expensive step on the Fisher Vector encoding through Equation 3 is to obtain posterior probabilities $\gamma_k(x_t)$, which represents the 80% of the encoding time. In order to have a stable implementation and at the same time, speed up this process, we optimize the operations involved on the expression of $\gamma_k(x_t)$ in the *log* domain (we have omitted the normalization term which is not relevant for this analysis)

$$\log \gamma_k(x_t) \propto \log(w_i) + \sum_{d=1}^{D} \left(x_t^d \log(\mu_{kd}) + (1-x_t^d) \log(1-\mu_{kd}) \right). \quad (6)$$

Furthermore, by taking advantage of the binary nature of the BRIEF descriptor, we propose an index-based data representation specifically tailored for speeding up the computation of $\log \gamma_k(x_t)$. Instead of having the binary vectors x_t and $1-x_t$, we store a vector of indices I_{x_t} and $I_{(1-x_t)}$ indicating the non-zero positions in x_t and in $1-x_t$ respectively (note that $I_{(1-x_t)}$ is the complement of I_{x_t} in the set $\{1, ..., D\}$). This will allow us to reduce the amount of multiplication operations by a factor of $2D$ and only compute the sum of $\log(\mu_{kd})$ for those $d \in I_{x_t}$ and the sum of $\log(1-\mu_{kd})$ for those $d \in I_{(1-x_t)}$,

$$\log \gamma_k(x_t) \propto \log(w_i) + \sum_{d \in I_{x_t}} log(\mu_{kd}) + \sum_{d \in I_{(1-x_t)}} \log(1-\mu_{kd}). \quad (7)$$

Besides, the denominator $\mu_{id}^{x_t^d}(1-\mu_{id})^{1-x_t^d}$ in Equation 3 can be expressed as $\mu_{id}x_t^d + (1-\mu_{id})\left(1-x_t^d\right)$, so the sets I_{x_t} and $I_{(1-x_t)}$ are also used to speed up its computation. The factor $F^{-1/2}$ does not depend on the samples and then it is computed only once.

III. Experimental results

This section presents an evaluation of the proposed video face descriptor BinVF^2. It is shown that our binary-based Fisher Vector representation achieves a significant speed-up compared to the SIFT-based Fisher Vector and other state-of-the-art methods, while obtaining comparable face verification accuracy results.

The experiments are conducted on the large scale database *YouTube Faces* (YTF) [6], which is considered the *de-facto* standard benchmark for automatic face verification in video. The dataset contains 3425 videos of 1595 subjects in unconstrained scenarios with significant variations on expression, illumination, pose, resolution and background. The length of videos is 181.3 frames on average. Two evaluation protocols are defined. The restricted protocol only uses for training the predefined same/not same labels, without considering the subject identity labels. The unrestricted protocol, on the other hand, allows training methods access to subject identity labels.

A. Face Verification effectiveness evaluation

In order to evaluate the effectiveness of BinVF^2 in the face verification task, we follow the restricted protocol of the YTF database [6] in which 5000 video pairs are specified, half of which are pairs of videos of the same person, and half of different people. These pairs are divided into 10 splits, each containing 250 same and 250 not-same pairs. Nine splits are used for training and the remaining split for testing. All the learning aspects involved in our method, including Bernoulli Mixture Models and the discriminative Fisher vector projections, were trained independently for each fold. The final performance is reported by the average accuracy obtained over the 10 splits. Three metrics: recognition rate (accuracy), area under curve (AUC) and equal error rate (EER) are used to report the face verification results.

First, we explore the effect of the number of components (K) of the BMM in the performance of our method. The

experiment was carried out on the unrestricted setting and the results, in terms of EER, are given in Figure 2. Comparing the obtained results for different values of K, it can be seen that the greater K the better the performance. However, for values of K greater than 512, little improvement is observed. As it can be appreciated in Equation 3 and 5, the value of K impacts directly in the computational cost of the FV encoding. Thus, in the rest of the experiments we use $K = 512$ as it provides the best trade-off between accuracy and efficiency. As in [10], all video frames were center cropped to 150×150, descriptors were computed densely over 24×24 patches using 2-pixel spacing at 5 levels of the scale-space pyramid with scale factor $\sqrt{(2)}$.

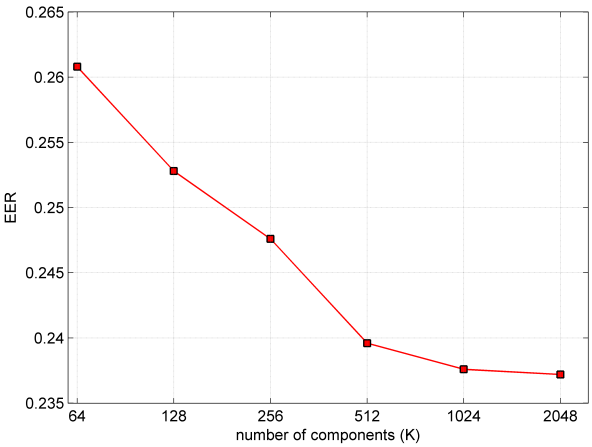

Fig. 2. Face verification performance (i.e., EER) for different values of K. For values greater than 512, the observed improvement is marginal.

In Table I, the performance of the proposed method and a comparison with state-of-the-art methods are presented. Deep learning based methods were not included in the comparison because they leveraged massive outside training data (millions) which is not covered by the restricted protocol used in this paper. Also, the ROC curves are depicted in Figure 3. As it can be seen the obtained accuracy is not the best one, however it is comparable with many of the state-of-the-art methods. It should be noticed that our main contribution is the efficient computation of video face representation without considerable accuracy lost. The following section focuses on evidencing the efficiency improvement obtained by our proposal.

B. Efficiency evaluation

There is not any defined protocol for evaluating the efficiency of video face recognition methods. In order to perform our computational time evaluation we report the average time for video face representation over all the videos in the restricted protocol of the YTF dataset. We compare the computational time of the BinVF^2 with that of VF^2 [10]. Additionally, we included the computation time of the feature extraction step of other state-of-the-art methods in the comparison.

Table II shows the total processing time of VF^2 and our BinVF^2, as well as the timing results of the feature extraction

Method	Accuracy ± SE	AUC	EER
MGBS (mean) LBP [6]	76.4 ± 1.8	82.6	25.3
MBGS + SVM [24]	78.9 ± 1.9	86.9	21.2
APEM_FUSION [25]	79.1 ± 1.5	86.6	21.4
STFRD + PMML [26]	79.5 ± 2.5	88.6	19.9
DDML(combined) [27]	82.3 ± 1.5	90.1	18.5
VSOF+OSS [5]	79.7 ± 1.8	89.4	20.0
VF^2 [10]	84.7 ± 1.4	93.0	14.9
EigenPEP [9]	84.8 ± 1.4	92.6	15.5
BinVF^2	79.8 ± 1.6	87.8	19.9

TABLE I
COMPARISON WITH STATE-OF-THE-ART RESULTS ON THE *YouTube Faces* DATABASE UNDER THE RESTRICTED PROTOCOL.

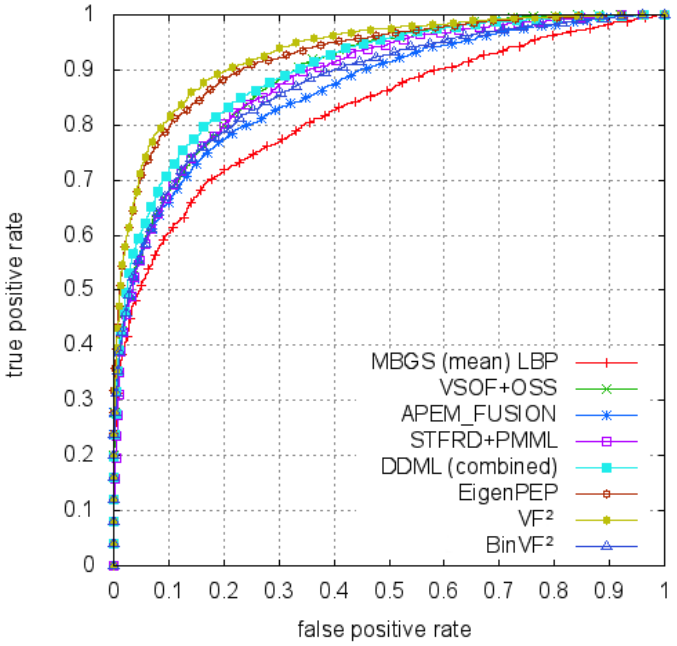

Fig. 3. (*best seen in color*). ROC curve on the *YouTube Faces* database under the restricted protocol.

stage. Since both implementations are able to run in multi-cores systems, we also include the computation time on a multi-core system. The reported times are expressed in seconds and the speed-up attained by our proposal is also given.

As it can be seen from Table II, for the single-core implementation, BinVF^2 significantly improved the computational time of the feature extraction step of VF^2, with a speed-up of 23x. For the multi-core implementation our method also outperformed the feature extraction stage of the VF^2 method, obtaining a speed-up of 27x. The whole system achieves a speed-up of 1.45x and 2.9x for the single-core and multi-core implementations, respectively. This computational time reduction is thanks to the use of an efficient binary descriptor as BRIEF and the reformulation of Equation 3, provided in Section II-C, which was possible due to the use of binary features.

BinVF^2 implementation was developed in C++. For the BRIEF descriptor we used the implementation provided in

	Single-core			Multi-core		
	VF^2 [10]	$BinVF^2$	Speed-up vs. VF^2	VF^2 [10]	$BinVF^2$	Speed-up vs. VF^2
Feature extraction	6.9	**0.3**	23x	5.4	**0.2**	27x
Feature extraction + FV encoding	19.9	**13.8**	1.45x	10.8	**3.7**	2.9x

TABLE II
COMPUTATION TIME COMPARISON WITH THE VF^2 ON THE *YouTube Faces* DATABASE USING THE SINGLE-CORE AND THE MULTI-CORE IMPLEMENTATIONS. ALL TIMES ARE EXPRESSED IN SECONDS AND REPRESENT THE AVERAGE COMPUTATION TIME FOR A VIDEO.

OpenCV 2.4.9. For the VF^2 method, we developed a C++ implementation using the VLFeat Library [28]. The VLFeat Library implements a fast version of dense SIFT which is 30 to 70 times faster compared to original SIFT, according to VLFeat authors. Note that we are already comparing our method in terms of efficiency with an optimized implementation.

For the case of the remaining state-of-the-art methods reported in Table I, the source codes are not publicly available. However we implemented, in C++, the feature extraction stage of some of these methods, which is usually the most expensive one, in order to compare them with our proposal. In Table III we compared the timing results of the feature extraction stage of our method with that of MBGS+SVM [24], VSOF+OSS [5], APEM_FUSION [25], EigenPEP [9] and VF^2 [10]. The parameters of the implemented methods were set as described by the authors in [5], [9], [10], [24], [25], respectively. It can be seen that $BinVF^2$ significantly improved the computational time of these methods, obtaining speed-ups ranging from 2.3x to 220x.

Method	Feature extraction computation time	Obtained speed-up
MBGS + SVM [24]	0.7	2.3x
APEM_FUSION [25]	20.1	67x
VSOF+OSS [5]	66.1	220x
EigenPEP [9]	3.5	11.6x
VF^2 [10]	6.9	23x
$BinVF^2$	**0.3**	-

TABLE III
COMPUTATION TIME COMPARISON OF THE FEATURE EXTRACTION STEP OF SEVERAL STATE-OF-THE-ART METHODS ON THE *YouTube Faces* DATABASE USING THE SINGLE-CORE IMPLEMENTATION. ALL TIMES ARE EXPRESSED IN SECONDS AND REPRESENT THE AVERAGE COMPUTATION TIME FOR A VIDEO.

All the experiments were conducted on a PC with an Intel i7-4770 CPU at 3.40 GHz and 8 GB of RAM.

C. Discussion

As it was shown, working with binary features allow us to simplify the computation of several steps of the FV representation; resulting in a considerable reduction of the computation time, a highly demanding characteristic in video face recognition. Although the current results in terms of accuracy do not outperformed the best ones from the state of the art, several improvements can still be made.

The process of encoding binary features in a Fisher Vector demands mathematical tools some times not available as easy as for real-valued data. A first example of that is the reduction of the dimension of local descriptors by PCA, which brings advantages not only from the computational point of view, but by reducing redundant information. Choosing a suitable method for reducing the dimension of binary descriptors is among our future work. Another very simple, but important aspect that can improve the discriminative power of our method, is the fact of incorporating the spatial information in the descriptor. Something so easy to achieve for descriptors lying in the reals, needs some elaboration for the case of binary descriptors in order to do not compromise the dimension and with that, the efficiency of the proposal.

A deeper aspect constitutes the statistical model used to approximate densities. (All the Fisher Vector encoding bases on the estimation of the density model assumed for the local descriptors.) In the case of SIFT descriptor, a GMM is used. Theoretically, it is known that GMM can approximate any smooth probability density by increasing the number of components. Nevertheless, a more appropriate model to approximate the distribution of binary features might exist, in particular for the case of BRIEF descriptor. All these aspects, some simpler than others, open a number of future works focused on improving the discriminative power of the proposed binary-based Fisher Vector encoding, incorporating elements already included in its counterparts like SIFT-based Fisher Vector encoding.

IV. CONCLUSION

In this paper we proposed to apply the Fisher Vector to the BRIEF binary features for video face recognition in order to reduce the computational effort of the entire representation. Binary features introduced a speed-up in the feature extraction phase, but also allowed the simplification of the Fisher Vector encoding since several multiplication operations were eliminated by using an index-based data representation. The conducted experiments validated the claimed contributions, obtaining a considerably more efficient video face representation with comparable accuracy performance with respect to state-of-the-art methods.

ACKNOWLEDGMENT

Author Noslen Hernández is supported by FAPERJ/CAPES (E45/2013). This work is supported in part by CONACYT Project No. 215546.

REFERENCES

[1] J. R. Barr, K. W. Bowyer, P. J. Flynn, and S. Biswas, "Face Recognition from Video: A Review," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 26, no. 5, 2012.

[2] L. Best-Rowden, B. Klare, J. Klontz, and A. Jain, "Video-to-video face matching: Establishing a baseline for unconstrained face recognition," in *IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, Sept 2013, pp. 1–8.

[3] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, 2014, pp. 1701–1708.

[4] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," *Proceedings of the British Machine Vision Conference*, vol. 1, no. 3, p. 6, 2015.

[5] H. M. Vazquez, Y. Martínez-Díaz, and Z. Chai, "Volume structured ordinal features with background similarity measure for video face recognition," in *International Conference on Biometrics, ICB 2013, 4-7 June, 2013, Madrid, Spain*, 2013, pp. 1–6. [Online]. Available: http://dx.doi.org/10.1109/ICB.2013.6612990

[6] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[7] J.-C. Chen, V. M. Patel, and R. Chellappa, "Landmark-based fisher vector representation for video-based face verification." in *ICIP*. IEEE, 2015, pp. 2705–2709.

[8] H. Li and G. Hua, "Hierarchical-pep model for real-world face recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2015, pp. 4055–4064.

[9] H. Li, G. Hua, X. Shen, Z. L. Lin, and J. Brandt, "Eigen-pep for video face recognition." in *ACCV (3)*, ser. Lecture Notes in Computer Science, D. Cremers, I. D. Reid, H. Saito, and M.-H. Yang, Eds., vol. 9005. Springer, 2014, pp. 17–33.

[10] O. M. Parkhi, K. Simonyan, A. Vedaldi, and A. Zisserman, "A compact and discriminative face track descriptor," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, 2014, pp. 1693–1700.

[11] H. Li, G. Hua, Z. Lin, J. Brandt, and J. Yang, "Probabilistic elastic matching for pose variant face verification," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3499–3506.

[12] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the fisher vector: Theory and practice," *International Journal of Computer Vision*, vol. 105, no. 3, pp. 222–245, 2013.

[13] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," *Lecture Notes in Computer Science : Computer Vision - ECCV 2004*, pp. 469–481, 2004.

[14] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[15] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Fisher Vector Faces in the Wild," in *British Machine Vision Conference*, 2013.

[16] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ser. ECCV'10, 2010, pp. 778–792.

[17] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: an efficient alternative to sift or surf," in *IEEE International Conference on Computer Vision*, 2011, pp. 2564–2571.

[18] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *IEEE International Conference on Computer Vision*. IEEE, 2011, pp. 2548–2555.

[19] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *IEEE Conference on Computer Vision and Pattern Recognition*. Ieee, 2012, pp. 510–517.

[20] Y. Uchida and S. Sakazawa, "Image retrieval with fisher vectors of binary features," in *2nd IAPR Asian Conference on Pattern Recognition*, 2013, pp. 23–28.

[21] A. Juan and E. Vidal, "Bernoulli mixture models for binary images," *17th International Conference on Pattern Recognition*, vol. 3, pp. 367–370, 2004.

[22] O. Miksik and K. Mikolajczyk, "Evaluation of local detectors and descriptors for fast feature matching," in *Pattern Recognition (ICPR), 2012 21st International Conference on*, Nov 2012, pp. 2681–2684.

[23] J. Lankinen, V. Kangas, and J.-K. Kamarainen, "A comparison of local feature detectors and descriptors for visual object categorization by intra-class repeatability and matching," in *21st International Conference on Pattern Recognition (ICPR)*. IEEE, 2012, pp. 780–783.

[24] L. Wolf and N. Levy, "The svm-minus similarity score for video face recognition," in *IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, 2013, pp. 3523–3530. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2013.452

[25] H. Li, G. Hua, Z. Lin, J. Brandt, and J. Yang, "Probabilistic elastic matching for pose variant face verification," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2013, pp. 3499–3506.

[26] Z. Cui, W. Li, D. Xu, S. Shan, and X. Chen, "Fusing robust face region descriptors via multiple metric learning for face recognition in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, 2013, pp. 3554–3561. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2013.456

[27] J. Hu, J. Lu, and Y.-P. Tan, "Discriminative deep metric learning for face verification in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2014, pp. 1875–1882.

[28] A. Vedaldi and B. Fulkerson, "Vlfeat: An open and portable library of computer vision algorithms," in *Proceedings of the 18th ACM International Conference on Multimedia*, ser. MM '10. New York, NY, USA: ACM, 2010, pp. 1469–1472. [Online]. Available: http://doi.acm.org/10.1145/1873951.1874249