Face Retrieval on Large-Scale Video Data

Christian Herrmann[1,2]
[1]Vision and Fusion Lab
Karlsruhe Institute of Technology KIT
Karlsruhe, Germany
christian.herrmann@iosb.fraunhofer.de

Jürgen Beyerer[2,1]
[2]Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB
Karlsruhe, Germany
juergen.beyerer@iosb.fraunhofer.de

***Abstract*—Increasingly large amounts of video data raise the question if large-scale face retrieval is feasible. To find fast and accurate matching strategies, an according face track descriptor is constructed by using local features, extended by an encoding of the respective measurement conditions. The feature encoding allows collecting all features of one face track together in a single feature set, where cumulative descriptors, known from image or object retrieval applications, especially bag of words and fisher vectors, can be applied. These descriptors are known to be viable for large-scale retrieval applications. To explore large-scale video face retrieval, we first evaluate on the largest available public datasets, i.e. YouTube Faces Database (YTF) and Face in Action Database (FiA). Finally, the behavior of face retrieval for increasing amounts of data is investigated by combining these datasets with 55K face tracks, collected from about 100 hours of TV data, making it the largest collection of face tracks we are aware of.**

***Index Terms*—face recognition, video retrieval, large-scale, fisher vector, bag of words**

I. Introduction

The growing amount of TV channels, video sharing websites or surveillance cameras provides masses of video data. To encounter these quantities of data, methods enabling automatic processing, structuring or information extraction are welcome. In most cases, the methods are supposed to understand the video in a similar way a human would, making it necessary to perform according tasks. Because humans especially focus on other humans in video, research addressing person detection, person recognition or action recognition is widely spread. This paper focuses on the challenge of *person retrieval* and especially the task to find further appearances of a given person in the video data by using the *face*. This allows the organization of the video data, the search for persons or forensic analysis if one thinks of surveillance data.

In this context, performing *face retrieval* refers to comparing video face representations which, in comparison to image based face retrieval, offer several face shots from consecutive frames to create a face representation. However, video data tend to have worse image quality and a less constrained environment. This leads to challenging variations in illumination, head pose or noise level which need to be addressed for a successful matching.

To process large amounts of data, a compact face descriptor is necessary, which is created by avoiding a frame based representation of the face track. Instead, we understand a face track as a set of measurements for the displayed face.

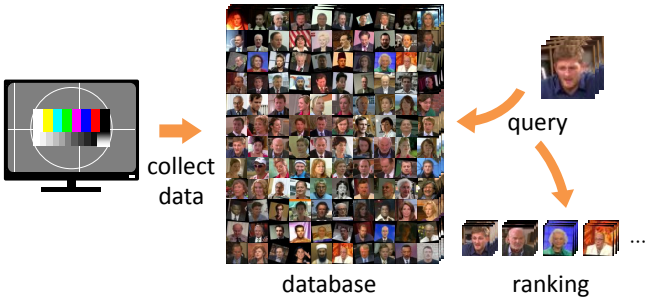


Fig. 1. Video face retrieval for large face databases.

Especially, it is unimportant in which specific frame, i.e. at what point in time, a measurement was performed, as long as the measurement conditions are known. Thus, by determining and encoding the respective measurement conditions, it is possible to combine all information from the track in a proper way. By understanding local features as measurements, we will show that augmenting the features with their respective location in the image and the head pose fully encodes the measurement conditions. The set of encoded features from one track is then compactly described by cumulative methods known from image or object retrieval tasks such as bag of visual words [1] or fisher vectors [2]. Especially, two strategies will be compared, the first being the baseline approach using bag of visual words with an inverted index [3]. Secondly, the recently presented VF^2 descriptor for face tracks [4] seems to be particularly compact and discriminative, thus, a similar strategy is applied.

Comparing several local features addresses illumination and noise issues. In addition to merely using the pixel intensity as feature, local binary patterns (LBP) [5] are employed because of their invariance to monotonic illumination changes. As a more noise resistant feature, in comparison to LBP, local directional patterns (LDP) [6] are evaluated and the final option are SIFT-features [7] following the suggestion of [4].

Current video based face recognition often focuses on small datasets, although the YTF dataset [8] is a step towards large-scale processing. Still, the official protocol includes only 5,000 track to track comparisons, allowing slow matching strategies. To give a clue about the speed of state-of-the-art face matching techniques, a comparison of the ones with the highest accuracy

on YTF[1] is presented in table I. Instead of face verification, we conduct face retrieval which requires many more track to track comparisons (almost 12M for YTF), especially if larger datasets will be addressed, where a lot of state-of-the-art methods seem infeasible.

The benefits and limitations of the proposed approach are explored on the largest publicly available datasets YTF [8] and FiA [9] in the first step, each having a size of about 3K face tracks. Performance evaluation as well as cross database evaluation is performed, which is especially interesting, because YTF is an in-the-wild dataset, whereas FiA was recorded in a constrained environment. For evaluation on a larger scale, we collected about 55,000 further in-the-wild face tracks from approximately 100 hours of TV program, which can be combined with the public data.

As first contribution, we show in section III that extending local features by the head pose and the image position uniquely encodes the entire recording situation, creating a theoretical foundation for this strategy. This allows the meaningful application of cumulative descriptors such as fisher vectors for the encoded features, presented in section IV. The proposed encoding significantly improves the retrieval result as is shown in section V. The final contribution is the collection of the hugest set of face tracks we are aware of, to examine the viability of the proposed approach in comparison to further approaches for large-scale face retrieval.

II. Related Work

Face retrieval. Because conventional image retrieval approaches, such as bag of words, are difficult to adapt to the face domain [3, 10], face retrieval remains a challenging task. This is mainly caused by the classical keypoint detection based descriptors such as SIFT, which have a tendency to fail for the smooth face surface. Thus, the usage of further information such as user feedback [11] or attribute based retrieval [12] seems to be in the focus of current face retrieval work. The problem of designing face retrieval systems which achieve a high performance and a small retrieval time seems to get little attention. Relevant work in that area includes small track descriptors by frame clustering [13] or fisher vectors [4], speeding up the distance measure [14] or applying cascade-based strategies [15, 16]. In particular, it is worth noting that some current top performing face verification systems seem inadequate for face retrieval, because computationally expensive classifier training steps are part of each face descriptor comparison [8, 17, 18, 19].

Face track description. Typically, face tracks are described in a frame based way, which means each frame is processed as a whole. Handling tracks based on the frame representations can be done by averaging on image [20], feature [21, 22] or decision level [8, 19, 23] over the whole track. Pairwise comparison and searching for the best match is another well performing approach [8], although it takes considerable time.

[1]http://www.cs.tau.ac.il/ wolf/ytfaces/results.html

TABLE I
Processing speed comparison of best performing methods on YTF. Because almost no paper gives numbers, most results were simulated or for the slow approaches roughly estimated. Simulation means using the exact descriptor structure and comparison method from the paper, only with random data.

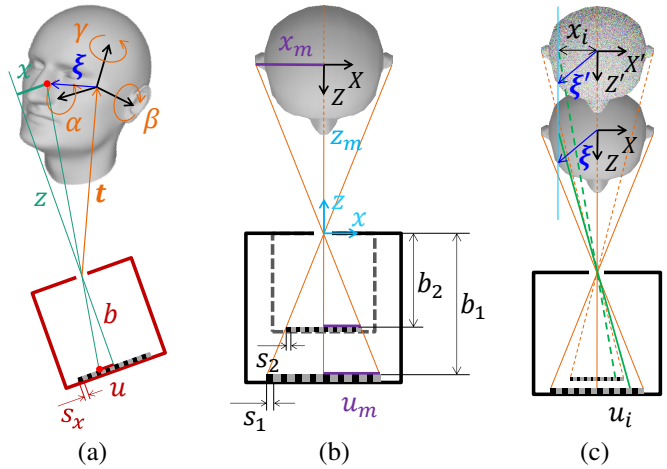
method	Accuracy ± SE	track comparisons per second	source
DeepFace [24]	91.4 ± 1.1	$4.8 \cdot 10^2$	simulated
VF^2 [4]	84.7 ± 1.4	$5.0 \cdot 10^5$	simulated
DDML [21]	82.3 ± 1.5	$5.0 \cdot 10^3$	paper
VSOF+OSS [31]	79.7 ± 1.8	$\sim 10^1$	estimated
STRFD+PMML [32]	79.5 ± 2.5	$< 10^1$	estimated
APEM-FUSION [18]	79.1 ± 1.5	$< 10^1$	estimated
MBGS [8]	76.4 ± 1.8	$1.5 \cdot 10^1$	public code

Randomly selecting only a few frames per track for comparison [24] reduces the computational effort. Instead of relying on single frames, one can model the space of the frames with a linear model [25, 26] or with a manifold [27, 28, 29]. Motivated by object retrieval work, a few approaches were presented recently, that avoid the frame based representation and instead use those based on local features [4, 18, 30]. Face tracks are represented by a set of local features which are collected across different frames. This allows to use cumulative descriptors such as bag of visual words [1] or fisher vectors [2], which enable the construction of large databases and performing fast queries. One key advantage of these cumulative descriptors is their constant size, independent of the track length. Consequently, our presented approach falls in this last category.

Feature augmentation. When collecting local features to describe an object, it was proven useful to augment the feature by its image coordinates [4, 18, 30], which means the concatenation of a feature vector and its image coordinates. We extend this location augmentation by a head pose augmentation and show that the combination of both uniquely encodes the measurement conditions of the respective local feature, thus creating a theoretical foundation for this previously heuristic strategy.

III. Encoding of local features

As argued before, a small track descriptor can be constructed out of a set of local features collected from the face track. However, one needs to keep in mind, that local features can originate from different positions in the face or represent different viewing angles caused by head motion and camera position. To keep this information and enable meaningful comparison of different local features, we encode the measurement conditions appropriately. Figure 2a gives an overview of the specifying parameters including head shape, position and camera parameters (pinhole camera), which need to be determined accordingly. First, a measurement targets a specific location $\boldsymbol{\xi} = (X, Y, Z)^T$ on the head/face and secondly, three rotation angles α, β, γ and a translation vector $\boldsymbol{t}$ describe the relative position of head and camera. Finally,



(a) (b) (c)

Fig. 2. Schematic drawings to visualize the recording situation: (a) all measurement parameters, including global target position (blue), head position (orange), local target position (green) and camera parameters (red), (b) virtual sensor orientation and its pixel size adaptation, (c) virtual sensor is invariant to object distance.

the intrinsic camera parameters of a pinhole camera are given by the distance b of the sensor from the projection center, the pixel scale $(s_x, s_y)^T$ and the sensor origin $(o_x, o_y)^T$. The camera calibration equations include the relations between all the parameters. Given the global coordinates $\boldsymbol{\xi} = (X, Y, Z)^T$ of a measurement, its camera coordinates are given by

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = R \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \boldsymbol{t}, \tag{1}$$

where $R = R_\alpha R_\beta R_\gamma$ denotes the rotation matrix. Using the camera coordinates $(x, y, z)^T$, the image coordinates $(u, v)^T$ are

$$\begin{pmatrix} u \\ v \end{pmatrix} = -\frac{b}{z} \begin{pmatrix} s_x^{-1} x \\ s_y^{-1} y \end{pmatrix} + \begin{pmatrix} o_x \\ o_y \end{pmatrix}. \tag{2}$$

The knowledge of the head pose, given by α, β and γ, and the image coordinates $(u, v)^T$ suffices to uniquely identify the target location $\boldsymbol{\xi}$ on the head in our special case:

Face detection and registration enforce that the origin of the global coordinate system lies on the z-axis of a virtual camera coordinate system, which means $\boldsymbol{t} = (0, 0, z_0)^T$. This defines a virtual camera, which is directly pointed at the head.

Face scaling to $p \times p$ pixels results in a unified pixel scaling factor $s = s_x = s_y$, constant sensor origin $o_x, o_y = const$, and because of face registration, to fixed image coordinates of the face boundary, denoted by $(u_m, v_m)^T = const$. This can be understood as taking the image by a virtual sensor with the resolution of $p \times p$ pixels which fits the scaling s exactly to match the size of the light rays of the face boundary, as shown in figure 2b.

A constant head size can be assumed in good approximation for all persons, at least for adults. According to [33] head width is $154 \pm 6mm$ and head height $199 \pm 7mm$ for Caucasian, and $158 \pm 7mm$ and $188 \pm 7mm$ for Chinese people, which shows that typical deviations are only in the order of a few percent and the assumption of a constant head size is feasible. Thus, the head width and height, and their respective halves x_m, y_m, are constant for each observation: $x_m, y_m = const$. Inserting half the width x_m into the u-coordinate part of equation 2 yields

$$u_m = \frac{b}{s z_m} x_m + o_x \tag{3}$$

and consequently

$$\frac{b}{s z_m} = \frac{u_m - o_x}{x_m} = const. \tag{4}$$

Of course, the same argumentation holds for using the y- and v-coordinates respectively. A constant value for $\frac{b}{s z_m}$ means that the virtual sensor also changes and fits to the size of the light rays of the face boundary, if the distance z_m between the face and the camera varies, which is illustrated by figure 2c.

To prove our claim that $(u, v, \alpha, \beta, \gamma)^T$ uniquely identifies $\boldsymbol{\xi}$, we need to show that for identical given head pose and image coordinates $(u, v, \alpha, \beta, \gamma)^T = (u', v', \alpha', \beta', \gamma')^T$, the respective face positions $\boldsymbol{\xi}$ and $\boldsymbol{\xi}'$ are identical. First, the following derivation shows that the camera coordinates in x- and y-direction are the same:

$$\begin{aligned}
\begin{pmatrix} x \\ y \end{pmatrix} &= -\frac{sz}{b} \cdot \begin{pmatrix} u - o_x \\ v - o_y \end{pmatrix} && (5)\\
&\approx -\frac{s z_m}{b} \cdot \begin{pmatrix} u - o_x \\ v - o_y \end{pmatrix} && (6)\\
&= -\frac{s' z'_m}{b'} \cdot \begin{pmatrix} u' - o'_x \\ v' - o'_y \end{pmatrix} && (7)\\
&\approx -\frac{s' z'}{b'} \cdot \begin{pmatrix} u' - o'_x \\ v' - o'_y \end{pmatrix} = \begin{pmatrix} x' \\ y' \end{pmatrix}. && (8)
\end{aligned}$$

Step 6 in the equation holds under the approximation that the distance between camera and head is significantly larger than the depth of the face $|z - z_m| \ll z$ and thus $z \approx z_m$. Step 7 uses the constant factor from equation 4. Given x and y, as well as the head orientation α, β and γ, the point $\boldsymbol{\xi}$ is uniquely identified on the head, if the assumption of a constant head size will be used again, which is illustrated in figure 2c for a known u_i.

Because the image position and the head pose uniquely encode the measurement conditions, we define an encoding vector $\boldsymbol{e}$ holding all position information: $\boldsymbol{e} = \tau \cdot (u, v, \alpha, \beta, \gamma)^T$, where τ is a scaling parameter which will be explained shortly.

Given a measurement vector $\boldsymbol{m}$ that describes the face under the encoded conditions $\boldsymbol{e}$, the encoded feature

$$\boldsymbol{w} = \begin{pmatrix} \boldsymbol{m} \\ \boldsymbol{e} \end{pmatrix} \tag{9}$$

allows uniform treatment of all measurements. Assuming normalized measurements with $||\boldsymbol{m}||_2 = 1$ the impact of the encoded position on the feature distance can be influenced by τ. By assuming euclidean distance, the encoding distance works additive to the squared measurement distance: $d^2(\boldsymbol{w_1}, \boldsymbol{w_2}) = d^2(\boldsymbol{m_1}, \boldsymbol{m_2}) + d^2(\boldsymbol{e_1}, \boldsymbol{e_2})$. And because $d(\boldsymbol{m_1}, \boldsymbol{m_2}) \leq 2$, caused by the normalization, the impact

of the encoding distance can be adjusted by τ. In this way, comparing measurements lying close to each other causes only a small penalty, none if the position is the same, while measurements denoting totally different parts of the face are significantly penalized. By ensuring this behavior, all encoded features of one track can be collected in a single feature set $W = \{\boldsymbol{w}_1, \dots, \boldsymbol{w}_n\}$ and the different feature locations are preserved.

Practical details. As measurements $\boldsymbol{m}$ we employ the local features LBP [5] (59 dimensions), LDP [6] (56 dimensions), SIFT [7] (128 dimensions) and simply the image intensity (64 dimensions). Each feature is determined from a local patch of the size 8×8 pixels, where the patch center is used for position encoding. For the histogram based LBP and LDP features, fusion of three different scales, corresponding to a filter size of 3, 4 and 5 pixels, is used [34]. In addition, as is recommended for histogram based descriptors [35], the Hellinger distance is used, which can be efficiently implemented in a preprocessing step with element-wise signed square rooting. Finally, PCA is applied to the feature vectors $\boldsymbol{m}$ and the dimension is reduced by half.

IV. Database representation of tracks

The feature set representation W of a track from the previous section has the advantage, that it can be used with typical retrieval algorithms. In addition to the baseline method, an inverted index with bag of visual words, which is widely used in object and image retrieval [1], we apply a face specific strategy using fisher vectors, similar to [4, 30].

Bag of visual words and inverted index. Briefly, clustering a domain specific training set and using the cluster centers as visual words generates a domain specific set of words (dictionary/codebook). A face track is described by the set (or bag) of included visual words, where nearest neighbor assignment of local features determines the according dictionary words. For the database, an inverted index is prepared, which contains for each dictionary word the database tracks this word occurs in. Querying the database with a new track is performed by determining the visual words for this track and looking these words up in the dictionary. The database tracks showing the highest number of matching visual words are ranked best. This method has the theoretical advantage, that the query time is independent of the database size, because a linear search is avoided.

Fisher vectors. The face domain is trained by fitting a gaussian mixture model (GMM) with K components $(\boldsymbol{\mu_k}, C_k, \alpha_k)$ having means $\boldsymbol{\mu_k}$, weights α_k and diagonal covariance matrices C_k. This is no restriction, because the previously applied PCA for the feature vectors includes a decorrelation. Fisher vectors capture the average differences of the first and second statistical moment for given face track features with respect to the GMM components, thus including additional information compared to a bag of words descriptor. Given the set of encoded local features W of one track, the fisher vector $\boldsymbol{\Phi}(W)$ is

$$\boldsymbol{\Phi}(W) = (\boldsymbol{a}_1, \dots, \boldsymbol{a}_K, \boldsymbol{b}_1, \dots, \boldsymbol{b}_K)^T \tag{10}$$

with

$$a_{k,i} = \frac{1}{n\sqrt{\alpha_k}} \cdot \sum_{j=1}^{n} p(k|\boldsymbol{w}_j) \frac{w_{j,i} - \mu_{k,i}}{\sigma_{k,i}} \tag{11}$$

and

$$b_{k,i} = \frac{1}{n\sqrt{2\alpha_k}} \cdot \sum_{j=1}^{n} p(k|\boldsymbol{w}_j) \left(\left(\frac{w_{j,i} - \mu_{k,i}}{\sigma_{k,i}} \right)^2 - 1 \right). \tag{12}$$

After L2-normalization and element wise signed square rooting [2] the normalized fisher vector will be denoted by $\boldsymbol{\phi}$. Following the suggestion of [30], a matrix $A \in \mathbb{R}^{p \times q}$, $p \ll q$ projects the normalized fisher vectors $\boldsymbol{\phi}$ to a low dimensional space, where the euclidean distance $d(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j) = ||A\boldsymbol{\phi}_i - A\boldsymbol{\phi}_j||_2$ is discriminative with respect to face recognition. The projection matrix A is trained according to [30]. A further option [36] is to learn joint projection matrices A and B in a way that the difference between the distance and kernel product can be used as a similarity score: $s(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j) = ||A\boldsymbol{\phi}_i - A\boldsymbol{\phi}_j||_2^2 - \boldsymbol{\phi}_i^T B^T B \boldsymbol{\phi}_j$. While it promises better recognition results, it doubles the necessary comparison time.

Matching time linearly depends on the size p of the low dimensional space in both cases, and also on the database size, because linear search is necessary. However, the size p of the target space can be chosen low enough to make this strategy applicable.

V. Experiments

Evaluation is performed with a 10-fold strategy, which means each dataset is randomly divided into 10 splits, whereof 9 splits are used as database and the tracks from the remaining split are used one-by-one as query tracks. In 10 repetitions, each of the 10 splits will be used as query track set. Altogether, this strategy results in equally many queries as the dataset has face tracks. To evaluate significant differences between methods, this amount of queries offers a larger statistical base compared to regular 10-fold cross-validation where only 10 samples are available. Motivated by [37], a randomization test is performed as statistical test on method pairs on an α-level of 0.05, which corresponds to approximately 2 standard deviations. Thus, results are presented as mean average precision map, which is the standard performance measure for retrieval evaluation, and it is stated where these values are significantly different according to the randomization test. The experiments are performed on a workstation having a six-core CPU with 3.4 GHz and 64 GB of memory.

A. Encoding

First, the potential benefit of the feature encoding is evaluated. For this purpose, the Multi-PIE [38] face image dataset is used because a large range of head poses is covered systematically in the data in contrast to public video face datasets.

TABLE II
EVALUATION OF FEATURE ENCODING AND ITS PARAMETERS. RANDOMIZATION TEST COLUMN DENOTES CASE NUMBERS WHICH YIELDED SIGNIFICANTLY WORSE RESULTS.

Case	Algorithm	K	p	Setting	map	Rand. test
1	fisher vec.	128	128	none	0.176	
2	fisher vec.	128	128	pose	0.194	
3	fisher vec.	128	128	spatial	0.420	1,2
4	fisher vec.	128	128	encoded	**0.472**	1,2,3
5	inv. index	64000			0.018	
6	inv. index	64000		encoded	**0.042**	5
7	fisher vec.	128	128	pose, $\tau = 1$	0.183	
8	fisher vec.	128	128	pose, $\tau = 2$	**0.194**	
9	fisher vec.	128	128	pose, $\tau = 4$	0.174	
10	fisher vec.	128	128	enc., $\tau = 1$	0.465	
11	fisher vec.	128	128	enc., $\tau = 2$	**0.472**	
12	fisher vec.	128	128	enc., $\tau = 4$	0.458	

Using the images from all 249 subjects from session 1 with all head poses under neutral illumination, the effectiveness of the feature encoding can be judged. Each image is handled like a face track consisting of one single frame leading to the same processing chain as for real face tracks consisting of multiple frames. The evaluation in this section, shown in table II, is limited to the encoding. Further variations and comparison to different approaches are examined in the next section with actual video face datasets.

Augmentation by pose, position and their combination (encoding) shows superior results compared to the baseline for both track representations: the fisher vectors in case 4 and the inverted index in case 6. In addition, it should be noted that the fisher vectors perform much better than the inverted index.

The scaling parameter τ is applied after normalization of position (range 0 to 1) and angle (range -1 to 1) values. Reasonable variations suggest that a value of $\tau = 2$ is optimal (cases 8,11), even though only insignificant differences are observed.

B. Public video datasets

Comparative evaluation is conducted on the public YouTube Faces Database (YTF) [8] and Face in Action Database (FiA) [9]. YTF consists of 3,425 face tracks from 1,595 different celebrities collected from YouTube, the average track length is 181 frames. FiA simulates an immigration act at the desk and is thus a dataset with a constrained environment and fixed camera positions. Nevertheless, it consists of 3,110 indoor tracks from 235 persons and the face tracks all share the same length of 200 frames. First, table III indicates the evaluation results of several parameter variations performed on YTF.

Local features are varied in cases 1 to 4 and it is shown that LBP performs best. The higher noise resistance of LDP seems to be useless on this data and the SIFT-features are also significantly outperformed. Thus, further experiments use LBP.

TABLE III
EVALUATION OF DIFFERENT FEATURES, PARAMETERS AND ALGORITHMS. RANDOMIZATION TEST COLUMN DENOTES CASE NUMBERS WHICH YIELDED SIGNIFICANTLY WORSE RESULTS.

Case	Algorithm	K	p	Setting	map	Rand. test
1	fisher vec.	512	64	Intensity	0.095	3,4
2	fisher vec.	512	64	LBP	**0.102**	3,4
3	fisher vec.	512	64	LDP	0.027	
4	fisher vec.	512	64	SIFT	0.064	3
5	fisher vec.	512	64		0.102	
6	fisher vec.	512	128		**0.109**	
7	fisher vec.	512	256		0.088	
8	fisher vec.	128	128		0.067	
9	fisher vec.	256	128		0.099	
10	fisher vec.	512	128		**0.109**	8
11	fisher vec.	512	128		0.109	
12	fisher vec.	512	128	encoded	0.154	11,13
13	fisher vec.	512	128	joint	0.124	
14	fisher vec.	512	128	j+e	***0.170***	11,12,13
15	inv. index	64000			0.013	
16	inv. index	64000		encoded	0.037	15
17	inv. index	128000		encoded	0.047	15
18	inv. index	256000		encoded	**0.061**	15,16,17

TABLE IV
EVALUATION RESULTS ON YTF AND FIA PUBLIC DATASETS, AS WELL AS A COMBINATION OF BOTH. SUPERSCRIPTS INDICATE RESULTS OF RANDOMIZATION TEST: A METHOD IS SIGNIFICANTLY BETTER THAN THE ONE INDICATED BY THE SUPERSCRIPT, INCLUDING EVERY WORSE ONE.

		map			*mean query time* in s		
No.	Method	YTF	FiA	comb.	YTF	FiA	comb.
1	NN	0.145^{2}	0.351^{4}	0.255^{4}	12.31	10.37	30.51
2	MSM	0.084^{3}	0.237^{3}	0.170^{3}	0.583	0.150	0.428
3	best shot	0.058	0.147	0.103	0.074	0.069	0.134
4	inv. index	0.061	0.290^{2}	0.183^{2}	0.100	0.103	0.114
5	fisher vec.	$\mathbf{0.170}^{1}$	$\mathbf{0.930}^{1}$	$\mathbf{0.552}^{1}$	**0.062**	**0.052**	**0.079**

The target dimension p of the low dimensional subspace for the fisher vector approach shows effects of overfitting if chosen too large (case 7).

The number of clusters K in the GMM used for domain adaptation is crucial for a good performance. Further increases beyond case 10 are impossible due to the memory capacity of our testing system.

Feature encoding and joint projection both increase the performance significantly and yield in combination the best overall result (case 14).

The inverted index approach cannot compete with the fisher vector method, probably caused by the lack of adaptability to the face domain. Here, K denotes the dictionary size, and again, feature encoding enhances the results significantly (compare cases 15,16).

The next evaluation step, shown in table IV, compares the results with three baseline face recognition methods which are fast enough to handle at least small scale retrieval tasks. First, the mutual subspace method (MSM) [26] which models the frame space by a linear subspace and allows fast track compar-

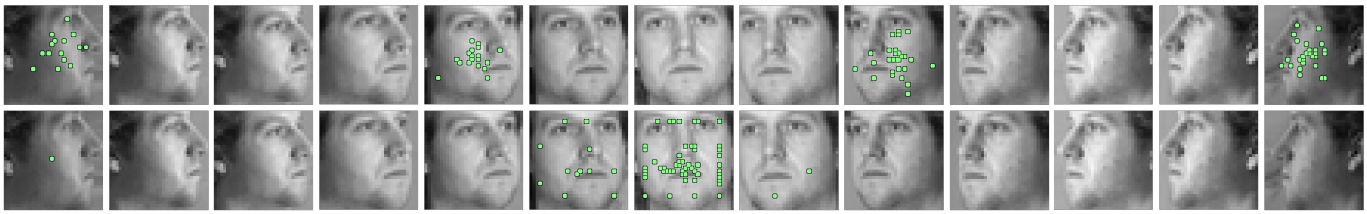


Fig. 4. Center locations of the 100 clusters with highest energy in the trained projection matrix W marked by dots. Top row shows training for FiA, bottom row for YTF (rounded to and shown on Multi-PIE head poses).

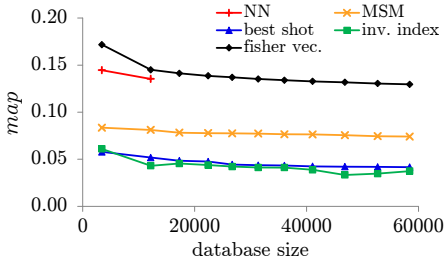


Fig. 5. Influence of database size on retrieval performance.

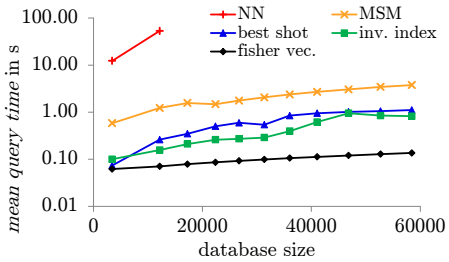


Fig. 6. Influence of database size on mean query time.

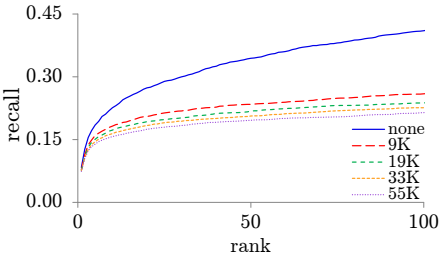


Fig. 7. Recall of ranks 1 to 100 for the fisher vector method and different numbers of distractors.

TABLE V
CROSS EVALUATION BETWEEN BOTH PUBLIC DATASETS.

Test	Training FiA	Training YTF
FiA	0.930	0.198
YTF	0.054	0.170

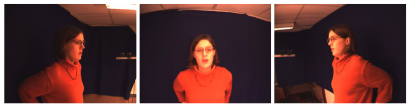


Fig. 3. Example data from FiA dataset, illustrating the three fixed camera positions.

ison because the principle angle is used as distance measure. Secondly, a best shot approach which selects the most frontal frame from each track for distance computation, which reduces the data size significantly and enables fast matching at the cost of accuracy. Finally, a nearest neighbor (NN) method which uses pair-wise frame descriptor comparison with the minimum pair-wise frame distance as track distance. This approach is the best performing one regarding conventional methods, as was shown by [8] (they called it ('min dist'), however, it is considerably slower than the previous two. All three methods use the frame-wise concatenated local patch LBP features as frame descriptor.

The mean query time denotes the time for one query. Note that N queries are performed during evaluation, if N denotes the database size. On YTF and FiA all methods except NN and MSM show query times below 100 milliseconds, but in the case of the best-shot and inverted index approach, this speed is bought by a significant reduction of matching accuracy. The face adapted fisher vector method, instead, achieves significantly better results than even the slow NN method, making it the best performing method, which can be explained by the training of the projection matrix that can be interpreted as a face domain adapted feature transformation, selection and weighting. The combination of both datasets shows approximately doubled query times, as is expected because of the doubled database size, with the exception of the inverted index method, where query time depends primarily on the dictionary size and database size has only a negligible influence.

The proposed fisher vector method increases the performance on YTF as well as FiA by a significant amount, however, the considerable improvement for FiA might be unexpected at first. A cross dataset evaluation further explores this effect: training on YTF and testing on FiA or vice versa shows, that these datasets have clearly different characteristics and the cross evaluation results in table V show a severe loss in performance. This is caused by the adaptation to the intrinsic challenges of the dataset during the training of the projection matrices. Especially, head pose variations, which are difficult to handle for the baseline methods, can be modeled and learned well by projection to subspaces, which was for example shown by partial-least-squares methods [39, 40]. Thus, training on data containing the same pose (camera) changes as the test data causes the considerable difference in the observed retrieval results for the FiA data, where only three fixed camera positions were used (figure 3).

Figure 4 shows the reason in the trained model: while YTF training (bottom row) focuses on frontal and almost frontal poses, the FiA trained model has the focus on profile and half-profile poses (top row). Especially the fact, that no frontal cluster is under the best ones for FiA might be unexpected, but can be explained as follows: there are 3 fixed camera positions to the left, front and right of the person's head (figure 3), and all persons move the head slightly during the video. Thus, each face track from the frontal camera includes some frames showing the head under minor pose variations, usually peaking at the pose where the trained clusters appear. Cross pose matching is easier for smaller pose changes, which is why the half-profile pose shots are selected, instead of frontal pose ones, for matching to the profile poses. This effect causes the observed problems in cross database evaluation, because YTF

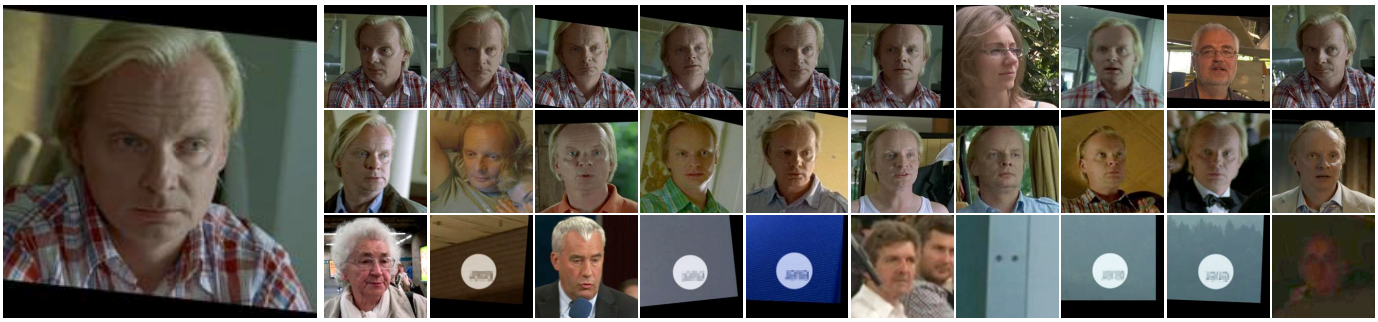


Fig. 8. Example of querying with TV data. Resulting ranking for one query track(large image): top row shows first 10 ranks, second row all correct matches between rank 11 and rank 50, and bottom row shows last 10 ranks.

contains almost no half-profile and profile shots.

C. Large-scale TV dataset

For large-scale evaluation, a set of face tracks is collected from about 100 hours of TV program. The collecting strategy is basically identical to the one which was used by [8] for YTF: A face tracker, based on the Viola-Jones face detector [41] and the Matlab multi object tracker are used, and only tracks longer than 30 frames are kept to exclude false detections. Faces are aligned by normalizing landmark positions [42] and afterwards converted to grayscale for feature computation. The dataset includes 55,020 face tracks with an average length of 81.6 frames, making it the largest collection of face tracks we are aware of.

The TV dataset tracks are used as distractors in the previously presented 10-fold evaluation strategy by adding them to the database tracks from the public YTF data. To have no undesired influence on the results, the TV face tracks must show different persons than the public dataset, which means care has to be taken to exclude the celebrities from YTF. We manage this problem by using the program from different local TV stations, focusing on local productions, especially excluding shows which potentially contain celebrities from YTF, such as news or hollywood movies.

First, adding more and more distractor tracks shows the influence of an increasing database size in figure 5. In doing so, a slight loss in performance can be observed. Baseline methods appear to encounter less performance degradation in relation to the fisher vector method which, nevertheless, remains the top performing method. Figure 6 shows that the query time increases linearly with the database size for all methods, now including the inverted index one, caused by the highly occupied index. All in all, the full evaluation requires about 200M track to track comparisons, thus the slowest baseline method NN can only be evaluated for the smallest set of distractors. All other methods are fast enough to handle databases at the given scale and fisher vectors are more than an order of magnitude faster than the baseline methods.

Finally, some tracks from the unannotated TV dataset are selected and used as query for a qualitative evaluation which shows reasonable results, as can be seen in figure 8 for one sample query. Top ranked results are tracks from the same scene, as expected, because illumination and head pose are very similar to the query. Further tracks from the same person are ranked below with a set of incorrect matches in between, however, these are few enough that a manual inspection of the ranking still makes sense. Finally, on the last ranks, clearly different faces and a set of false face detections can be found. Especially, the logo of one particular tv show seems to confuse the face tracker, but our retrieval algorithm proves robust to this problem by ranking these tracks at the bottom.

VI. Conclusion

It is shown that augmentation of local features with their image position and the head pose uniquely describes the recording situation of the face. This leads to a theoretical justification to apply cumulative descriptors which are widely used for retrieval tasks. Consequently, two compact face track descriptors based on bag of words and fisher vectors are built out of a set of local features, which are collected all over the face track. While experiments for the bag of words descriptor with an inverted index show fast, but low quality retrieval results, the fisher vector approach achieves significantly better results than the baseline methods, while being more than one order of magnitude faster caused by better domain adaptation. The results show that video face retrieval is still incapable of achieving the high performance known from some image retrieval tasks where matching samples can reliably be found in databases of millions of images. However, the presented results are at a level where manual inspection allows significant benefits in video analysis.

VII. Acknowledgment

This study was partially supported by the German Federal Ministry of Education and Research (BMBF) as part of the MisPel program under grant no. 13N12063.

References

[1] J. Sivic and A. Zisserman, “Video Google: A text retrieval approach to object matching in videos,” in *International Conference on Computer Vision*, 2003, pp. 1470–1477.

[2] F. Perronnin, J. Sánchez, and T. Mensink, “Improving the fisher kernel for large-scale image classification,” in *European Conference on Computer Vision*. Springer, 2010, pp. 143–156.

[3] C. Herrmann and J. Beyerer, “Fast Face Recognition by Using an Inverted Index,” in *Proc. SPIE 9405, Image Processing: Machine Vision Applications VIII*, 2015.

[4] O. M. Parkhi, K. Simonyan, A. Vedaldi, and A. Zisserman, "A Compact and Discriminative Face Track Descriptor," in *Computer Vision and Pattern Recognition*, 2014.

[5] T. Ahonen, A. Hadid, and M. Pietikainen, "Face Description with Local Binary Patterns: Application to Face Recognition," *Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.

[6] T. Jabid, M. H. Kabir, and O. Chae, "Local directional pattern (LDP) for face recognition," in *Consumer Electronics*, 2010, pp. 329–330.

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

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

[9] R. Goh, L. Liu, X. Liu, and T. Chen, "The CMU Face In Action (FIA) Database," *Analysis and Modelling of Faces and Gestures*, pp. 255–263, 2005.

[10] Z. Wu, Q. Ke, J. Sun, and H.-Y. Shum, "Scalable face image retrieval with identity-based quantization and multireference reranking," *Pattern Analysis and Machine Intelligence*, vol. 33, no. 10, pp. 1991–2001, 2011.

[11] B. M. Smith, S. Zhu, and L. Zhang, "Face image retrieval by shape manipulation," in *Computer Vision and Pattern Recognition*, 2011.

[12] B. Chen, Y. Chen, Y. Kuo, and W. Hsu, "Scalable face image retrieval using attribute-enhanced sparse codewords," *Transactions on Multimedia*, vol. 15, no. 5, pp. 1163–1173, 2013.

[13] M. Zhao, J. Yagnik, H. Adam, and D. Bau, "Large Scale Learning and Recognition Of Faces in Web Videos," in *Automatic Face and Gesture Recognition*, 2008, pp. 1–7.

[14] C. Huang, S. Zhu, and K. Yu, "Large Scale Strongly Supervised Ensemble Metric Learning, with Applications to Face Verification and Retrieval," *NEC Technical Report*, 2011.

[15] C. Herrmann and J. Beyerer, "Maximizing Face Recognition Performance for Video Data Under Time Constraints by Using a Cascade," in *Advanced Video and Signal Based Surveillance (AVSS), 2014 11th IEEE International Conference on*. IEEE, 2014, pp. 181–186.

[16] ——, "Pyramid Mean Representation of Image Sequences for Fast Face Retrieval in Unconstrained Video Data," in *Lecture Notes in Computer Science 8888, Advances in Visual Computing, 10th International Symposium, ISVC 2014, Proceedings, Part II*, 2014, pp. 304–314.

[17] T. Berg and P. N. Belhumeur, "Tom-vs-Pete Classifiers and Identity-Preserving Alignment for Face Verification." in *British Machine Vision Conference*, 2012.

[18] H. Li, G. Hua, Z. Lin, J. Brandt, and J. Yang, "Probabilistic elastic matching for pose variant face verification," in *Computer Vision and Pattern Recognition*, 2013.

[19] L. Wolf and N. Levy, "The svm-minus similarity score for video face recognition," in *Computer Vision and Pattern Recognition*, 2013.

[20] R. Jenkins and A. Burton, "100% Accuracy In Automatic Face Recognition," *Science*, vol. 319, no. 5862, pp. 435–435, 2008.

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

[22] E. G. Ortiz, A. Wright, and M. Shah, "Face recognition in movie trailers via mean sequence sparse representation-based classification," in *Computer Vision and Pattern Recognition*, 2013.

[23] M. Tapaswi, M. Bäuml, and R. Stiefelhagen, ""Knock! Knock! Who is it?" probabilistic person identification in TV-series," in *Computer Vision and Pattern Recognition*, 2012, pp. 2658–2665.

[24] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Computer Vision and Pattern Recognition*, 2014, pp. 1701–1708.

[25] H. Cevikalp and B. Triggs, "Face recognition based on image sets," in *Computer Vision and Pattern Recognition*, 2010.

[26] O. Yamaguchi, K. Fukui, and K. Maeda, "Face Recognition Using Temporal Image Sequence," in *Automatic Face and Gesture Recognition*, 1998.

[27] O. Arandjelović and R. Cipolla, "A pose-wise linear illumination manifold model for face recognition using video," *Computer Vision and Image Understanding*, vol. 113, no. 1, pp. 113–125, 2009.

[28] K. Lee, J. Ho, M. Yang, and D. Kriegman, "Video-Based Face Recognition Using Probabilistic Appearance Manifolds," *Computer Vision and Pattern Recognition*, vol. 1, pp. 313–320, 2003.

[29] M. E. Wibowo, D. Tjondronegoro, L. Zhang, and I. Himawan, "Heteroscedastic probabilistic linear discriminant analysis for manifold learning in video-based face recognition," in *Workshop on Applications of Computer Vision*, 2013, pp. 46–52.

[30] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Fisher vector faces in the wild," in *British Machine Vision Conference*, vol. 1, no. 2, 2013, p. 7.

[31] H. Mendez-Vazquez, Y. Martinez-Diaz, and Z. Chai, "Volume structured ordinal features with background similarity measure for video face recognition," in *International Conference on Biometrics*, 2013.

[32] 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 *Computer Vision and Pattern Recognition*, 2013.

[33] R. Ball, C. Shu, P. Xi, M. Rioux, Y. Luximon, and J. Molenbroek, "A comparison between Chinese and Caucasian head shapes," *Applied Ergonomics*, vol. 41, no. 6, pp. 832–839, 2010.

[34] C. Herrmann, "Extending a local matching face recognition approach to low-resolution video," in *Advanced Video and Signal Based Surveillance*, 2013.

[35] L. Wolf, T. Hassner, and Y. Taigman, "Descriptor based methods in the wild," in *ECCV Workshop on Faces in Real-Life Images: Detection, Alignment, and Recognition*, 2008.

[36] D. Chen, X. Cao, F. Wen, and J. Sun, "Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification," in *Computer Vision and Pattern Recognition*, 2013, pp. 3025–3032.

[37] M. D. Smucker, J. Allan, and B. Carterette, "A comparison of statistical significance tests for information retrieval evaluation," in *Information and Knowledge Management*, 2007.

[38] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-PIE," *Image and Vision Computing*, vol. 28, no. 5, pp. 807–813, 2010.

[39] M. Fischer, H. K. Ekenel, and R. Stiefelhagen, "Analysis of partial least squares for pose-invariant face recognition," in *Biometrics: Theory, Applications and Systems*, 2012, pp. 331–338.

[40] A. Li, S. Shan, X. Chen, and W. Gao, "Cross-pose face recognition based on partial least squares," *Pattern Recognition Letters*, 2011.

[41] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[42] M. Everingham, J. Sivic, and A. Zisserman, "Hello! My name is... Buffy–Automatic naming of characters in TV video," in *British Machine Vision Conference*, 2006.