Temporally Subsampled Detection for Accurate and Efficient Face Tracking and Diarization

Nam Le*†, Alexander Heili*, Di Wu* and Jean-Marc Odobez*†
*Idiap Research Institute, Martigny, Switzerland
†Ecole Polytechnique Federal de Lausanne, Switzerland
Email: { nle, aheili, dwu, odobez }@idiap.ch

***Abstract*—Face diarization, i.e. face tracking and clustering within video documents, is useful and important for video indexing and fast browsing but it is also a difficult and time consuming task. In this paper, we address the tracking aspect and propose a novel algorithm with two main contributions. First, we propose an approach that leverages state-of-the-art deformable part-based model (DPM) face detector with a multi-cue discriminant tracking-by-detection framework that relies on automatically learned long-term time-interval sensitive association costs specific to each document type. Secondly to improve performance, we propose an explicit false alarm removal step at the track level to efficiently filter out wrong detections (and resulting tracks). Altogether, the method is able to skip frames, i.e. process only 3 to 4 frames per second - thus cutting down computational cost - while performing better than state-of-the-art methods as evaluated on three public benchmarks from different context including a movie and broadcast data.**

I. Introduction

Nowadays, a large amount of multimedia data like news, debates, talkshows, documentaries or series is being produced and broadcast through multiple TV or internet channels. To increase the value of these data and the experience of users (curators, journalists, any viewer), making the content more easily accessible through search, retrieval and fast browsing tools is needed. Such tools should not only be based on scarce metadata like titles or dates but should benefit from semantic contents available inside each video. In particular, as people are often of high interest for users, segmenting videos into homogeneous segments according to human identity has been the topic of important research [11], [5], [8], [19].

To address the task of identifying people in videos, faces must be localized and connected into face tracks within a shot, which is important as it has been shown that using tracks leads to better face representation than individual images [15]. Due to the wide range of media content and amount of videos, this has two main challenges that we investigate in this paper for the two above tasks: robustness and computational cost.

To obtain face tracks, typical diarization systems [8], [11] rely on frontal face detectors like the Viola-Jones (VJ) detector [20] due to availability and speed. Then, for tracking, Kanade-Lucas-Tomasi interest point trackers (KLT) are often used to link detections and associate them over time [11], [15], [14], [19], [17]. Nevertheless, given the diversity of image backgrounds and faces that can appear in challenging illumination and poses, the detector may miss detections and produce a large amount of false ones. To counter this lack of robustness, systems usually only use the frontal face detector (thus missing a large amount of near profile faces), apply it at every frame to obtain better detection statistics, and complement it with forward/backward tracking or complex per track skin filtering procedure [8] to remove false alarms. Although much better detectors exist, they are usually not used due to their expensive running time in normal hardware. In addition, even with fast detectors, due to the very large amount of data to be processed, being able to cut down the computational time is desired.

In this paper, we propose a novel tracking approach that takes advantage of the state-of-the-art multi-view DPM detector within a fast tracking method to benefit from the detector accuracy without the expense of running time. More precisely, we use a fast version [2] of the DPM detector. For tracking, we rely on the tracking-by-detection framework of [6] initially proposed for human tracking in surveillance context with static cameras, and extend it in several ways to the multi-face tracking domain in media data by adding new similarity features and a more advanced false track removal step. Thus results in a tracker that exploits time-interval sensitive discriminative multi-cue appearance and motion association costs learned in an unsupervised way, allowing an easy adaptation to each media document type. In particular, since long term connectivity between detections is exploited, to the contrary of most frame-to-frame methods, our approach delivers competitive results while only having to process 3 to 4 frames per second.

Extensive experiments on three public datasets (including movie and broadcast data) demonstrate the benefit of the different contributions.

The next Section reviews existing works complementary to ours. Section III describes the tracking framework. Section IV presents the conducted experiments to support our propositions. Finally, Section V concludes the paper with further discussion and future works.

II. Related Work

Our overall goal is face diarization, which is mainly composed of 3 different parts: detection, tracking, and face clustering. However, in this paper, we are mainly in the design of efficient methods for the two first steps, detection and tracking, and comment on them below in the context of our task.

Face detection. This is the bottleneck that greatly influences the performance of the whole system. Missing detections can be caused by profile faces, lighting conditions, or intrinsic variability of faces. Meanwhile, background with detailed textures can be easily mistaken for real faces, which creates noise for tracking as well as clustering. Such problems not only diminish the utility of the system but can also annoy practical users. Most diarization systems rely on the standard VJ detector [20] which has shown to have low accuracy on competitive datasets such as PASCAL faces or FDDB [13]. To improve this, 2 strategies are proposed: aggregating multiple detectors to increase the recall rate [14] and filtering with upper body detectors to increase the precision rate [19]. Both strategies slow down the system significantly. In another direction, deep neural networks has achieved high accuracy on face detection problem. However, such networks require GPU and considerable trade-off in accuracy [22], even if dedicated architecture and specialized hardware may increase their speed in the future. Therefore in our work, we rely on the DPM detector [4], which is highly competitive and easy to integrate without side effects [13].

Face tracking. From the detections, tracking aims to create a set of continuous face tracks. KLT [10] is commonly applied due to its speed and simplicity [14], [19]. However, this tracker is sensitive to long occlusion and drifting over time. On the other hand, tracks can also be obtained by associating detections. Often, tracklets are first formed, eg based on time, motion, and color [21] or location, size, and pose [17], and used as base units for further linking by optimizing a graphical model using similar features [21] or using discriminatively trained face appearance models based on DCT statistics [17].

All the aforementioned systems require the detector to be applied every frame to create face tracklets and tracks reliably. On the contrary, we show that by extending the framework of [6], we can benefit from long-term connectivity, with parameters estimated in an unsupervised fashion and thus adapted to each type of data source (news, movies, TV series, etc.), and do not need feature tracking thanks to the use of instant visual motion information. Altogether, this allows speeding up the tracker by applying the detector sparsely.

III. Face detection and tracking system

Our system comprises of three main stages: detection, tracking, and false track removal. We describe each stage with a focus on the features and optimization of the tracking process.

A. Face detection

We employ the multi-view DPM model, which achieved state-of-the-art results [4], [13]. However, due to the numerous convolutions required, a main disadvantage of DPM is its computational cost, which can take up to 3s/frame for HD videos. Thus, we use a sped-up variant leveraging Fourier transforms to accelerate the processing [2]. Furthermore, as shown in the experiments, thanks to the increased accuracy w.r.t. the VJ detector, we only need to apply the face detector 3 to 4 times per second, which considerably decreases the computational cost for detection.

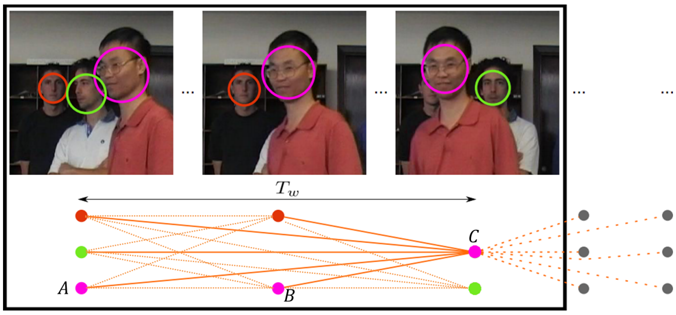

Fig. 1. Tracking as graph clustering task. The detections form the nodes, and a long-term connectivity is used, i.e. all links between pairs of nodes within a temporal window T_w are used to define the cost function. Long-term connectivity combined with time-interval sensitive discriminative pairwise models and visual motion enables dealing with missed detections, e.g. due to occlusion, as well as skipped frames.

B. Face tracking overview

We propose to leverage and extend the multi-human tracking method proposed in [7], [6], by adding new features, handling sparse detections over time (Section III-D), and adding a false track removal step (Section III-E).

Our approach is illustrated in Figure 1. Face tracking is formulated as a labeling problem within a Conditional Random Field (CRF) framework. Given the set of face detections $Y = \{y_i\}_{i=1:N_y}$, where N_y is the total number of detections, we search for the set of corresponding labels $L = \{l_i\}_{i=1:N_y}$ such that faces belonging to the same identity are assigned the same label. This is done by optimizing the posterior probability $p(L|Y, \lambda)$, where λ denotes the set of model parameters. Under some assumption, this is equivalent to minimizing the following energy potential:

$$U(L) = \left(\sum_{(i,j)\in\mathcal{V}} \sum_{r=1}^{N_s} w_{ij}^r \, \beta_{ij}^r \, \delta(l_i - l_j) \right), \tag{1}$$

with the Potts coefficients defined as:

$$\beta_{ij}^r = \log \left[\frac{p(S_r(y_i, y_j)|H_0, \lambda_{\mathbf{\Delta}_{ij}}^r)}{p(S_r(y_i, y_j)|H_1, \lambda_{\mathbf{\Delta}_{ij}}^r)} \right]. \tag{2}$$

with the different terms defined as follows. First, the energy involves N_s feature functions $S_r(y_i, y_j)$ measuring a similarity between detection pairs as well as confidence weights w_{ij}^r for each detection pair. Importantly, note that a *long-term connectivity* is exploited, in which the set of valid pairs $\mathcal{V}$ contains *all pairs* whose temporal distance $\mathbf{\Delta}_{ij} = |t_j - t_i|$ is lower than T_w, where T_w is usually between 1 and 2 seconds. This contrasts with most frame-to-frame tracking or path optimization approaches. For instance, in Fig. 1, even if there is a path from A to B and B to C for the same track, the link A to C is also exploited in the cost function, resulting in better conditioned objective function.

Secondly, the Potts coefficients themselves are defined as the likelihood ratio of the probability of feature distances under two hypotheses: H_0 if $l_i \neq l_j$ (i.e. detections do not belong to the same face), or H_1 when labels are the same. In practice, this allows to *incorporate discrimination*, by

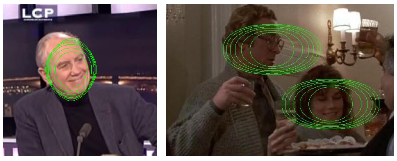

Fig. 2. Position. The different iso-contours of value 0 of the Potts costs for different values of $\boldsymbol{\Delta}$ (i.e. location of detections occurring after $\boldsymbol{\Delta}$ frames around each shown detection and for which $\beta = 0$), learned in an unsupervised fashion from TV REPERE (left) and Hannah (right). In the region delimited by a curve, association will be favored, whereas outside it will be disfavored. Curves show that more motion is expected on the Hannah movie, than on the TV data.

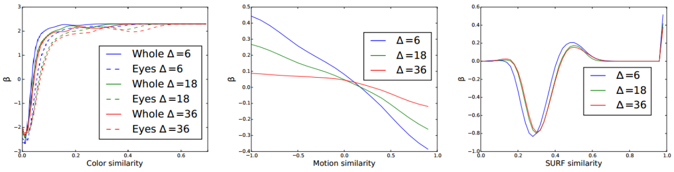

Fig. 3. Automatically learned Potts functions β for different similarity functions and some $\boldsymbol{\Delta}$ values. Left: color cue. Middle: motion. Right: SURF.

quantifying how much features are similar and dissimilar under the two hypotheses, and not only on how much they are similar for the same identity as done in traditional path optimization of many graph-based tracking methods. Furthermore, note that as these costs depend on the set of parameters $\lambda^r_{\boldsymbol{\Delta}_{ij}}$, they are *time-interval sensitive*, in that they depend on the time difference $\boldsymbol{\Delta}_{ij}$ between the detections. This allows a fine modeling of the problem and will be illustrated below.

Finally, in Eq. 1, $\delta(.)$ denotes the Kronecker function ($\delta(a) = 1$ if $a = 0$, $\delta(a) = 0$ otherwise). Therefore, coefficients β^r_{ij} are only counted when the labels are the same. They can thus be considered as “costs” for associating or not a detection pair within the same track. When $\beta^r_{ij} < 0$, the pair of detections should be associated so as to minimize the energy 1, whereas when $\beta^r_{ij} > 0$, it should not.

C. Features and association cost definition

Our approach relies on the unsupervised learning of time sensitive association costs for $N_s = 8$ different features. Below, we briefly motivate and introduce the chosen features and their corresponding distributions. We illustrate them by showing the Potts curves (for their learning see next section), emphasizing the effect of time-interval sensitivity and their easy adaptation to different datasets.

Position. The similarity is the Euclidean distance $S_1(y_i, y_j) = \mathbf{x}_i - \mathbf{x}_j$, with $\mathbf{x}_i$ the image location of the i^{th} detection y_i. The distributions of this feature are modeled as zero mean Gaussians whose covariance $\Sigma^H_{\boldsymbol{\Delta}}$ depends on the hypothesis (H_0 or H_1) and the time gap $\boldsymbol{\Delta}$ between two detections. Fig. 2 illustrates the learned models by plotting the zero iso-curves of the resulting β functions. We can notice the non-linearity with respect to increasing time gaps $\boldsymbol{\Delta}$ (curves are closer and closer as $\boldsymbol{\Delta}$ increases), and the difference between document types: more static heads are expected in the REPERE TV programs than in the Hannah movie.

Motion cues. Motion similarity between detection pairs is assessed by comparing their relative displacement and their visual motion. The similarity is computed as the cosine of the angle between these two vectors. Intuitively, if a face moves in a constant direction, the displacement between its detections and their visual motion will be aligned, leading to a motion similarity close to 1, whereas for unrelated faces, this would be more random. Note that the use of such an *instantaneous motion information* differs from frame-to-frame KLT tracking and *is not affected by occlusion or drift*. The resulting β curves in the middle plot of Fig. 3 confirm the above intuition, but surpringly indicate that this motion information is more discriminative for short time intervals. Indeed, in the TV data, when considereing 1 to 2 seconds time intervals, head motion might be less reliable as people are more likely to shake their heads back and forth, leading to flatter β curves.

Appearance (color). Faces are represented by multi-level color histograms in 4 different regions: the whole face, and the mouth, eye, and nose regions. The similarity between histograms of the same region of the detections is measured using the Bhattacharyya distance D_h, and the distributions of this distance is modeled using a non-parametric method. Example of Potts curve β are shown in Fig. 3, Left. We can notice here that the statistics associated to each region are relatively different, and although we would not expect so, also varies with the time gap $\boldsymbol{\Delta}$ between detections.

Appearance (SURF). Color is sufficient to discriminate between faces. We thus propose to exploit SURF [1] descriptors computed at interest points detected within the face bounding box as more structured appearance measures. As similarity measure, we use the average Euclidean distances between pairs of nearest keypoint descriptors from the two detections. We model the distributions of the similarity measures with a non-parametric approach. As can be seen in the right plot of Fig. 3, the Potts coefficient β is negative for a SURF similarity around 0.3, thus encouraging association for such values. On the other hand, positive coefficients for larger distances - around 0.5 - discourage the association.

D. Parameter learning, optimization

Given our non-parametric and time interval sensitive cost model, the number of parameters in λ is quite large. We adopt an unsupervised learning strategy to estimate λ directly from data, removing the need for tedious track annotations. Learning is done in two steps. First, we rely on a simple assumption that up to a short term interval, pairs of closest and second closest detections come from the same person or not, respectively. This allows us to learn model parameters under each of the two hypotheses, and perform a first round of tracking. Second, we use the resulting tracks to refine and obtain the model parameters up to larger time intervals. Note that although on test data we are only interested in the parameters at multiples of $\boldsymbol{\Delta}_{sk}$ (the frequency at which face detections is applied), during training tracking is done using all intervals to obtain reliable tracks for the parameter refinement.

Optimization. For computational efficiency, we used a sliding window algorithm that labels the detections in the current frame as the continuation of a previous track or the creation

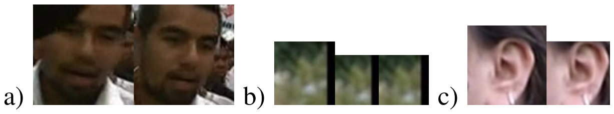

Fig. 4. False alarm removal examples. a) Short but positive track falsely removed by [6] but kept by our model. b) Negative track correctly removed mainly thanks to image position and detection size. c) Negative track falsely kept by [3] due to skin color but correctly removed by our model.

of a new one, using an optimal Hungarian association algorithm relying on all the pairwise links to the already labeled detections in the past T_w instants. A second step (Block ICM) is then conducted, which allows reasoning at global level by swapping track fragments at each time instant [6].

E. False alarm track removal

The CRF provides face tracks, some of which may correspond to false alarms. In other trackers [6], [14], these are often simply removed based on track length. On broadcast data, this is not sufficient given the content diversity and track length limited by shot duration. Thus, in contrast to [6], we further learnt a classifier to filter the false alarm tracks based on more cues [18]. For each face track, motion, position, size, and detection confidence scores were collected and accumulated to form a feature vector. Then, a linear SVM classifier was trained to distinguish true tracks from false ones.

To train our model, we created a training database by annotating 9364 face tracks from 9 videos from the development set of the REPERE corpus, and used the obtained model to conducted our evaluation on other datasets as reported in the experiment section. The linear SVM model achieves 93.3% cross validation accuracy. Based on the weights, the most important cues are detection score, width, and position. Fig. 4 illustrate qualitatively why this multi-cue model is superior to false alarm models based on single feature such as duration [6] or skin filtering [3]. Furthermore, our linear model with simple features is fast to work with large video corpus.

IV. Experiments

A. Experimental protocol

Our primary interest is in the handling of brodcast data. However, to benchmark the results and allow comparison, we used the three following datasets:

- "Frontal and Turning" consists of 2 videos recorded with a fixed camera [12], each involving 4 subjects moving around with frequent occlusions and fast movements (Frontal video) or many profile faces (Turning video).
- "Hannah" corresponds to the movie "Hannah and her sisters" by W. Allen and was fully annotated [14]. It is challenging due to moving cameras and faces of many characters at multiple poses and angles.
- REPERE. It features 9 programs including news, debates, and talk shows from two French TV channels, along with sparse annotations available through the REPERE challenge [5]. From the Test 2 subset of this challenge, we randomly selected 27 videos equally from each program, covering approximately 18 hours of data. Nine videos were used for parameter learning (tracker, FAR step, cf sections III.D and E), and the 18 other ones for testing.

	Frontal				Turning			
T_w-$\mathbf{\Delta}_{sk}$	PH	MT	Frag	IDS	PH	MT	Frag	IDS
36-1	23	**6**	**16**	**0**	15	**4**	**9**	**0**
36-6	23	5	18	**0**	18	3	16	**0**
36-12	35	2	30	1	30	0	27	**0**
48-1	21	**6**	**15**	**0**	14	**4**	**8**	**0**
48-6	24	5	17	**0**	16	**4**	14	**0**
48-12	33	3	30	2	30	1	26	1
[17]	11	4	24	13	11	2	**8**	4
[21]	15	5	25	10	15	**4**	**8**	5

TABLE I
Tracking results on "Frontal and Turning". Parameters denote: T_w, up-to how many frames apart are pairwise links built. $\mathbf{\Delta}_{sk}$: detections are only extracted every $\mathbf{\Delta}_{sk}$ frames.

As performance measures, we relied on standard metrics available in the papers used for comparison.

B. Tracking evaluation

Frontal/Turning dataset. We used the metrics of [9] used by [21]. Mostly Tracked (MT, number of groundtruth trajectories correctly tracked for more than 80% of their duration), Fragmentation (Frag, number of times groundtruth trajectories are interrupted, the smaller the better), ID Switches (IDS, number of times tracked trajectories change matched groundtruth identity). Results are reported in Table I for different parameter configurations: tracking window size T_w and frequency $\mathbf{\Delta}_{sk}$ at which detection is performed (every $\mathbf{\Delta}_{sk}$ frames).

Compared to other methods which typically do first short term tracklet creation and then tracklet linking [17], [21], our system outperforms them in both scenarios, with much less ID switches and Frag overall, and higher or the same MT. Indeed, as tracking association relies on a longer temporal window and the detection recall is enough, we can track most of the groundtruth tracks. Importantly, the number of IDS is minimized (0), which is crucial for further person clustering and naming in our target application. On the other hand, there are a bit more PH in our case, esp. in the Frontal sequence, since as our method provides more coverage of the tracks (higher MT), it also produces shorter segments than [17], [21] which are not merged with the main track. However, these fragmented tracks could be further joined through a further face clustering step [11], [8], [19].

Parameters T_w and $\mathbf{\Delta}_{sk}$ variations. One can observe that with longer tracking context T_w (compare results for $T_w = 48$ vs $T_w = 36$), tracks are more likely to recover from temporary occlusions or missed detections, which usually results in less Frag and higher MT, but the difference is not large. On the other hand, when detector is applied very scarcely (e.g. $\mathbf{\Delta}_{sk} = 12$), we observe a noticeable performance decrease (e.g. 2 vs 6 Mostly tracked people for T_w on the Frontal sequence). Fig. 5 shows the only example of IDS in Frontal sequence, which is due to a combination of adverse circumstances. However, applying the detection every $\mathbf{\Delta}_{sk} = 6$ frames produces only a small loss of performance (except for Frag in Turning sequence, which can be recovered by face clustering), and

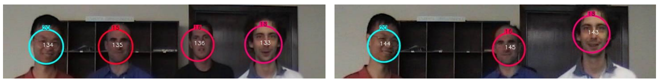

Fig. 5. The only ID switch observed with the 31-12 configuration. The two images are separated by exactly 12 frames, and unfortunately, the person corresponding to the T15 tracker moves towards the person T16 and perfectly occludes him in the second image. Hence, as they both share relatively similar skin color, both the color and motion features are not discriminant, whereas the positional association and the absence of alternative detection for T16 favors the labeling of detection 145 as T16 rather than T15.

since detection is one of the computation bottlenecks, provides a good trade-off between performance and speed.

Hannah dataset. Frame by frame annotation allows us to evaluate the method with both detection-based as well as track-based metrics, as used by [14]:

- Frame-based: comparison of faces returned by the system (track boxes) and groundtruth faces (GT boxes) results in 3 measures: False Positive (FP), False Negative (FN), and Multiple Track (MultT, ratio of GT boxes with multiple track matches). It is important to note that these boxes are considered **after** the tracking phase.
- Track-based evaluation reflects the purity of matching through 3 metrics: Tracker Purity (TPu), Object Purity (OPu), and Purity. TPu is the average of purity of all tracks, where the track purity is defined as the ratio of frames for which the track box correctly identifies the GT track box it is associated with, over the output track length. Similarly, the object purity of each GT track is defined as the ratio of frames for which it is correctly identified by the output track it is associated with, over the total length of the GT track. Purity measures the overall quality of face tracks based on TPu and OPu.

Systems. We compare our method with 2 strong baselines, each of them illustrating a different approach to the problem. In the first one [14], the detector is a combination of frontal and profile Viola & Jones (VJ) detectors with Zhu and Ramanan multi-pose detector [23], which produces high frame-based score. Tracking is done with an improved version of the KLT tracker, and track removal is based on duration. The second baseline utilizes only the frontal VJ detector and per track adapted GMM-based skin filtering [3]. Tracking is done by associating detection pairs using SURF matching similarity together with forward/backward search. For our systems, there are 4 different configurations, all with $T_w = 36$: $\mathbf{\Delta}_{sk} = 1$ and $\mathbf{\Delta}_{sk} = 6$ both without or with false alarm removal (FAR) (trained on REPERE, cf. Section III.E).

Analysis. Results are shown in Table II. When looking at our system, we note that when $\mathbf{\Delta}_{sk} = 1$, there are more detections, thus a lower FN but a high FP. The application of the FAR classifier significantly decreases the number of FP, with almost no change in FN. The high MultT can be explained partly because our method is generating longer tracks (in this cases interpolated frames near occlusion generate extra matches with GT faces), and partly because the DPM detector sometimes outputs multiple detections of slightly different scale for the same face, resulting in some spurious short track duplicates. Interestingly, results with $\mathbf{\Delta}_{sk} = 6$ show that combining skipping frames and FAR can lead to even more precise result than just FAR alone (lower FP and MultT) in complex settings such as movies, at the cost of a lower recall (higher FN). Compared to other methods, we can note that because skin-filtering minimizes false alarms and frontal faces are easier to connect with exhaustive search, the baseline [3] produces fewer false positives and a high tracker purity score than the [14], but at the cost of a much lower recall (larger FN and smaller OPu). Although [14] relies on multiple detectors applied at every frame, our tracker with $\mathbf{\Delta}_{sk} = 6$ results in much better performance: 3 times less FP, for similar FN. At the track-based level, our system outperform the baselines. Because false alarm tracks are taken into account when computing the tracker purity TPu, FAR significantly contributes to improving TPu (moving from 56.3% to 91.1%) with a very minor drop of OPu.

REPERE corpus. This corpus does not provide dense annotation but only one head position at a single reference frame and the temporal bounds when it appears for each given track. Therefore, we can only evaluate the performance indirectly by measuring the detection performance on the reference frames[1]. Based on the intersection of the groundtruth polygons and tracked hypotheses in these frames, one can calculate the recall, precision, and F1-measure. To report performance at the track-level, we weight the detection errors by the track duration. This results in the false alarm time rate (denoted FA. Time), i.e. the sum of the false alarms weighted by the track duration divided by the total duration of all reported tracks. We can similarly compute the Missed time (Missed T.). In the 27 selected videos, there are 4130 annotated heads.

Systems. Our tracker configuration is $\mathbf{\Delta}_{sk} = 6$ with a $T_w = 36$ temporal connectivity. The baseline is [3], and consists of frontal detector, skin filtering, and SURF-based tracker as described in the previous experiment.

Analysis. Results are shown in Table III. At frame level, thanks to the multi-view DPM face detector, the recall is increased by quite a large margin (on the test set, from 43.8% for the baseline relying on VJ detector to 58.2%). However, this is at the cost of an increase of false alarms (reduction of precision). Again, it should be reminded that besides applying detection at every frame, the baseline [3] also employs GMM-based skin filtering, which explains their high precision. Our result can be improved by applying the false alarm track removal step. In that case, the precision increases by almost 13% while only losing 1% in recall. Compared to the baseline, we gain more than 12% in F1-measure. Looking at the weighted measure, we can note that though our system without FAR has around 20% false detections at frame level, the total FA time is only around 10%. This means that false alarms still tend to form short tracks on average. Finally, Fig. 6 illustrates errors produced by the system.

Computational performance. Besides accuracy, the running time is a major concern for face diarization systems. On an

[1] Note that the tracker output in these frames may have been generated through interpolation of the obtained tracks.

	Frame-based			Track-based		
	FP (%)	FN (%)	MultT (%)	OPu (%)	TPu (%)	Purity (%)
Ours, $\boldsymbol{\Delta}_{sk} = 1$, no FAR	36.3	**33.7**	2.5	35.9	54.2	43.2
Ours, $\boldsymbol{\Delta}_{sk} = 1$, FAR	10.2	33.9	2.28	**36.4**	87.3	**50.2**
Ours, $\boldsymbol{\Delta}_{sk} = 6$, no FAR	29.6	40.6	6.08	28.1	56.3	37.5
Ours, $\boldsymbol{\Delta}_{sk} = 6$, FAR	**5.3**	42.6	0.72	27.5	**91.1**	42.3
[14]	17.4	39.2	0.39	22.5	50.6	31.2
[3]	13.2	66.6	**0.07**	12.3	66.7	20.7

TABLE II

HANNAH DATASET. EVALUATION OF OUR TRACKING FRAMEWORK AGAINST OTHER BASELINE SYSTEMS.

	Dev set					Test set				
	Recall	Precision	F1	Missed Time	FA Time	Recall	Precision	F1	Missed Time	FA Time
Baseline [3]	39.2	**94.3**	55.4	68.1	5	43.9	**96.8**	60.3	62.7	8.9
Ours, no FAR	**60.6**	79.1	68.6	52.0	8.7	**58.2**	82.2	68.2	53.5	11.8
Ours, with FAR	59.1	93.1	**72.3**	52.5	4	57.0	94.8	**71.2**	55.6	7.3

TABLE III

TRACKING PERFORMANCE ON THE 27 VIDEOS FROM THE REPERE DATASET, SPLIT ACCORDING TO OUR TRAINING/TEST SETS (SEE SECTION 4.1).

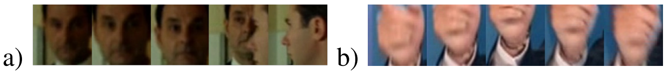

Fig. 6. Two typical failure examples on REPERE. a) Switch caused by two faces crossing each other too slowly. b) False alarms due to hands looking similar to real face tracks.

Intel(R) Core(TM) i7-4930K CPU @ 3.40GHz machine, for HD images (1024x756), the detector can process 3-4 frames/s, yielding real time speed when applying it only on 4 frames per second. For comparison, frontal and profile VJ detectors run at 6 - 7 frames/s on the same machine. For 1 hour of HD video, the tracker costs around 1.5 hour in total including motion estimation and detections.

V. CONCLUSION AND FUTURE WORKS

We presented our detection and tracking system in the context of face diarization. Unlike others, our method takes advantage of the robust multiview DPM detector and of the tracker ability to well exploit long term connectivity to perform robust tracking even under low detection frame rate. The method also benefit from a supervised false alarm removal model based on different cues. Our contributions evaluated on standard datasets yield state-of-the-art results with a substantial decrease in computational load. The diversity of the used datasets (short sequences, movies, large broadcast corpus) demonstrates the potential of our system to be exploited for large-scaled indexing and retrieval systems.

Future works include using the pose information as additional cue in the tracking framework, eg. with [16], as well as further investigation of the face track clustering step to reach clean face diarization.

Acknowledgement This research was supported by the European Union project EUMSSI (FP7-611057).

REFERENCES

[1] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *ECCV*, pages 404–417. Springer, 2006.

[2] C. Dubout and F. Fleuret. Exact acceleration of linear object detectors. In *ECCV*, pages 301–311. Springer, 2012.

[3] E. El Khoury. *Unsupervised video indexing based on audiovisual characterization of persons*. PhD thesis, Toulouse University, 2010.

[4] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 32(9):1627–1645, 2010.

[5] A. Giraudel, M. Carré, V. Mapelli, J. Kahn, O. Galibert, and L. Quintard. The repere corpus: a multimodal corpus for person recognition. In *LREC*, 2012.

[6] A. Heili, A. Lopez-Mendez, and J. M. Odobez. Exploiting Long-Term Connectivity and Visual Motion in CRF-based Multi-Person Tracking. *IEEE Trans. on Image Processing*, 2014.

[7] A. Heili and J. M. Odobez. Parameter estimation and contextual adaptation for a multi-object tracking crf model. In *IEEE Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, pages 14–21, Jan 2013.

[8] E. Khoury, P. Gay, and J.-M. Odobez. Fusing Matching and Biometric Similarity Measures for Face Diarization in Video. In *ACM ICMR*, 2013.

[9] Y. Li, C. Huang, and R. Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *CVPR*, 2009.

[10] B. D. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*, 1981.

[11] M. Everingham, J. Sivic, and A. Zisserman. Taking the bite out of automated naming of characters in TV video. *Image and Vision Computing*, pages 545–559, 2009.

[12] E. Maggio, E. Piccardo, C. Regazzoni, and A. Cavallaro. Particle phd filtering for multi-target visual tracking. In *ICASSP*, 2007.

[13] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool. Face detection without bells and whistles. In *ECCV*, 2014.

[14] A. Ozerov, J.-R. Vigouroux, L. Chevallier, and P. Perez. On evaluating face tracks in movies. In *ICIP*, 2013.

[15] O. M. Parkhi, K. Simonyan, A. Vedaldi, and A. Zisserman. A compact and discriminative face track descriptor. In *CVPR*, 2014.

[16] E. Ricci and J. Odobez. Learning large margin likelihood for realtime head pose tracking. In *IEEE ICIP*, 2009.

[17] M. Roth, M. Bauml, R. Nevatia, and R. Stiefelhagen. Robust multi-pose face tracking by multi-stage tracklet association. In *ICPR*, 2012.

[18] M. Tapaswi, C. Corez, M. Bauml, H. Ekenel, and R. Stiefelhagen. Cleaning up after a face tracker: False positive removal. In *ICIP*, 2014.

[19] M. Tapaswi, O. M. Parkhi, E. Rahtu, E. Sommerlade, R. Stiefelhagen, and A. Zisserman. Total cluster: A person agnostic clustering method for broadcast videos. In *ICVGIP*, page 7. ACM, 2014.

[20] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001.

[21] B. Wu, S. Lyu, B.-G. Hu, and Q. Ji. Simultaneous clustering and tracklet linking for multi-face tracking in videos. In *ICCV*, 2013.

[22] S. Yang, P. Luo, C.-C. Loy, and X. Tang. From facial parts responses to face detection: A deep learning approach. In *ICCV*, 2015.

[23] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, 2012.