ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Facial Descriptors for Identity-Preserving Multiple People Tracking

Michalis Zervos[1] (michail.zervos@epfl.ch)

Horesh Ben Shitrit[1] (horesh.benshitrit@epfl.ch)

François Fleuret[⋆] (francois.fleuret@idiap.ch)

Pascal Fua (pascal.fua@epfl.ch)

CVLab, School of Computer and Communication Sciences
Swiss Federal Institute of Technology, Lausanne (EPFL)

EPFL-REPORT-187534
July 2013

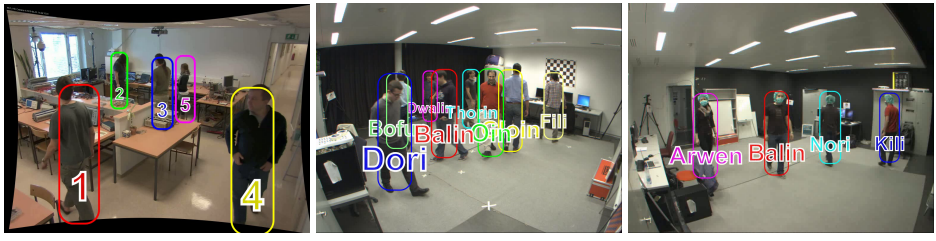
Abstract. In this report, we show that facial descriptors can be used very effectively in conjunction with a tracklet-based multi-person tracker both to localize and to identify or re-identify people over long sequences. Thus, we can reliably deliver both trajectories and identities in crowded scenes. Furthermore, the whole approach is fast enough to be implemented in real-time. Our key insight is that this can be done even though the faces can only be recognized relatively infrequently.

[1] Both authors have contributed equally to this work
[⋆] IDIAP Research Institute, Switzerland
This work was supported in part by the DACH project

1 Introduction

Multiple people tracking algorithms tend to rely on image appearance remaining relatively constant over time to produce trajectories corresponding to specific individuals. This is especially true of those trackers that depend on matching appearance attributes from frame-to-frame. However, commonly-used features, such as soft biometrics—height, body proportions, or gait—or color histograms, are either hard to measure in natural environments or cannot be relied upon for identity preservation over long periods of time. This is especially true when people move in and out of the field of view.

By contrast, facial features are highly distinctive and tend to remain consistent over time. It would therefore seem natural to use them for this purpose and face-recognition technology should be an important component of people tracking systems. Remarkably, it is not, probably because in video sequences such as those of Fig. 1, only a small fraction of the faces can be recognized.



(a) (b) (c)

Fig. 1: Representative results on multi-camera sequences. (a) Without *a priori* knowledge of people's identity or appearance, we can reliably localize people and assign a unique identifier to each one, even when they leave and come back. (b) With such knowledge, we can keep track of people's actual names. (c) This works even when people's heads are partially covered by surgical masks and hats. The corresponding videos are supplied as supplementary material. To preserve anonymity, the names have been intentionally changed to characters' names from the "The Hobbit" and "The Lord of the Rings".

In this paper, we aim to change this by showing that facial features can nevertheless be used to achieve accurate and efficient people tracking with reliable identity labeling. The challenge we address is to effectively leverage sparse facial information. We define individual appearance models which are used to identify and re-identify people in temporally distant frames and prevent our tracking algorithm from switching people's identities.

To this end, we exploit the ability of many current people trackers [1–4] to track individuals over short trajectory segments, often referred to as *tracklets*, without knowledge of their identities. Since it is much more likely that a useful identification will be made within a tracklet than in a single temporal frame, even sparse identification data becomes useful. This is key because, even though face identification algorithms [5–7] can achieve high success rates when image-resolution is high enough and people look

directly at the camera, their performance drops drastically when people are not actively cooperating. In such an uncontrolled setting, they only provide useful data once in a while.

More specifically, we start from a recently proposed Multi-Commodity Network Flow approach of linking tracklets [3, 8]. It is well suited to our needs because it can exploit the kind of sparse appearance information discussed above, but it requires appearance models for individuals expected to appear in the scene to be given *a priori*, which is impractical for surveillance-like applications. We overcome this limitation by computing face descriptors [9] wherever a face can be detected and associated to a specific tracklet, clustering these descriptors, and using the resulting clusters to provide the required appearance models. We use sequences such as those depicted by Fig. 1 and our supplementary material to demonstrate that this lets us reliably group tracklets corresponding to the same person and, consequently, re-identify people leaving and re-entering the visible area.

Our main contribution is a novel algorithm, which integrates a state-of-the-art tracking procedure with appearance measures derived from face recognition techniques, in a unified probabilistic framework. It efficiently produces complete and reliable trajectories over long sequences in addition to identifying the people in the scene.

2 Related Work

In this section, we first discuss the need for appearance-based models for the purpose of reliably tracking multiple people. We then briefly discuss the current state of this technology and justify the specific choice we made of a face-recognition algorithm to test our ideas.

2.1 People Tracking

Early approaches to finding people in images largely focused on frame-to-frame tracking, which involves predicting the people's location in a frame given an estimate in the previous one. The emphasis has now shifted to tracking-by-detection in which people are detected in individual frames and the detections are then linked across time. This prevents drift and provides robustness to occasional failures.

Most state-of-the-art approaches follow this tracking-by-detection paradigm and operate on graphs whose nodes can either be those where a detector has fired [10, 11] or tracklets, that is, short temporal sequences of consecutive detections that are very likely to correspond to the same person [4, 8, 12–16]. On average, they are much more robust than the earlier tracking methods but require appearance-based models to guarantee that only individual detections or tracklets corresponding to the same individuals end up being linked. These appearance-based models typically rely on color, texture, or rough shape of different parts of people's bodies. However, none of these can be measured very consistently, especially when people come in close proximity which is when the models are most needed.

Face descriptors have been used to make monocular head tracking more robust [17, 18]. The resulting algorithms can recover from long face occlusions or people exiting

and re-entering the field of view. However, they cannot provide 3D trajectories of multiple people while preserving their identities, which is what we propose and demonstrate in the paper.

2.2 Face Recognition

Image-based face recognition has been extensively researched over the past 20 years and is usually implemented as a two-step process, first detection [19] and then recognition [20, 21].

For detection purposes, the well-known Viola-Jones approach [22] remains a leading contender. It relies on Haar-like features computed at multiple scales and locations and used to classify image patches as containing a face or not. A variation of this algorithm was proposed in [23]. It relies on Binary Brightness Features (BBF) that are computed at individual pixel locations making it faster than Viola-Jones while achieving comparable performance. More recently a unified framework [24] for multi-view face detection, pose estimation and landmark localization was proposed, that advanced the state-of-the-art on several standard benchmarks, on the cost of computational time.

Early approaches to face recognition tended to model the face as a whole [25, 26]. Because they use all the available image information and take geometry into account, they are effective for aligned faces under controlled conditions but are not robust to illumination, pose, and facial expression changes. More recent methods designed to handle these difficulties rely on first extracting local features and then employing a classifier for identifying the subject. The Local Binary Pattern (LBP) Histograms approach [9], which encodes the gray level information of small image regions into features and then takes into account the spatial configuration of the resulting feature vectors, is an attempt at getting the best of both worlds. It is often used as an component of state-of-the-art face recognition methods [7, 27].

In our work, we use an LBP-based face recognition algorithm, because its performance is sufficiently close to the state-of-the-art for our tracking purposes, while allowing for real-time implementation.

3 Tracking with Sparse Facial Information

As discussed in § 2.1, one of the most promising approaches to identity-preserving people tracking in uncontrolled environments is to first compute short trajectory segments, or tracklets, that unambiguously describe the motion of a single individual and then to group them into complete trajectories. However, to fulfill their full potential, such approaches require appearance-based measures that can be used to unambiguously link all tracklets corresponding to the same person. To acquire those tracklets, we use our Linear Programming approach [28] to create complete trajectories that may include some identity switches but can easily be segmented into single-individual tracklets [8], such as those of Fig. 2.

In this section, we first summarize the approach used to create the tracklets. We then discuss how to create appearance models based on facial features, which are very reliable but at the cost of being exploitable only at distant time intervals. Finally, we

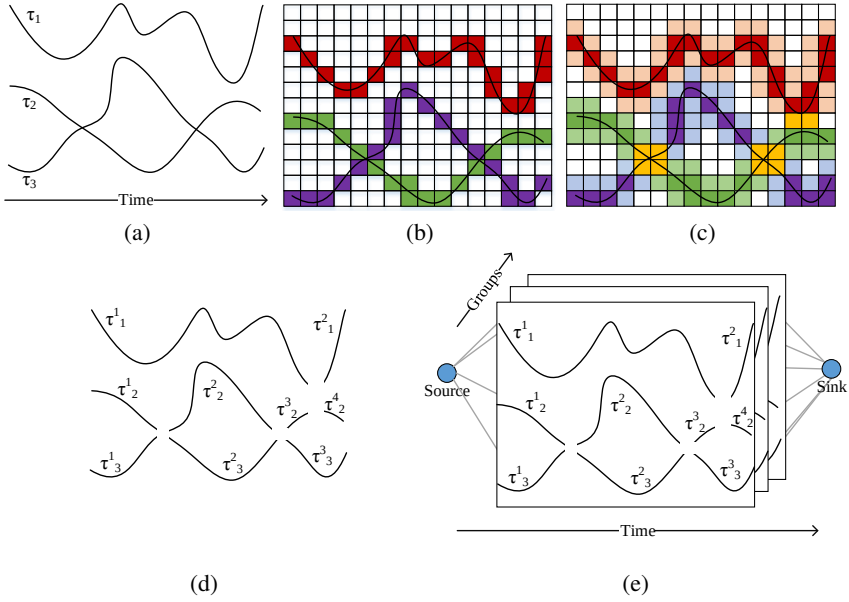


Fig. 2: Splitting trajectories into single-identity tracklets. (a) Three trajectories that may include identity-switches. (b) Each one is a set of vertices occupied at successive time instants and we assign a different color to each. (c) Grid cells within a distance of 1 from the trajectories are shown in a color similar to that of the closest trajectory but less saturated. The vertices that are close to more than one trajectory appear in yellow and are those that could be used to connect them. (d) The yellow vertices are used as splitting points to produce tracklets. Note that two trajectories do not necessarily have to cross to be split; it is enough that they come close to each other. (e) The resulting tracklets become the nodes of multi-layer Directed Acyclic Graph.

show how to use them to compute the appearance-based probabilities, that the tracking algorithm depends on to reliably group tracklets.

3.1 Creating the Tracklets

Given image sequences acquired by multiple synchronized cameras such as those of Fig. 1, we customize our publicly available software [28, 29] to generate trajectories on a discretized grid by solving a Linear Program, as depicted by Fig. 2(b). This algorithm reliably finds all people but completely ignores appearance information. Thus, the trajectories often include identity switches, especially at locations where two of them come close to each other. As suggested in [8], we handle this by finding all the locations that could be used to connect one trajectory to the other, such as the yellow grid cells of Fig. 2(c). In practice, these cells are the only places where an identity switch could occur and we therefore take the trajectory fragments connecting them to be our tracklets.

Assuming the number L of identifiable people present in the scene to be known *a priori*, we duplicate each tracklet $L+1$ times so that tracklet τ^l stands for subject l following the corresponding trajectory, for $0 \leq l \leq L$, where $l=0$ denotes unidentified people. To link back these tracklets into complete identity-preserving trajectories, we first build a directed acyclic graph (DAG) whose nodes are the tracklets and edges represent potential connections between tracklets that share an endpoint. To each edge is associated a flow $f_{i,j}^l$ that can be either zero or one to indicate if a person l went from τ_i^l to τ_j^l. The tracklets are also connected to a source and sink to allow people to enter and leave the area of interest, which results in the full graph depicted by Fig. 2(e). Note that there are no edges connecting tracklets τ_i^l to τ_j^k if $l \neq k$ because identity switches are not allowed along trajectories.

Given this DAG, the maximum a posteriori probability flows $f_{i,j}^l$ are those that maximize

$$\sum_{l,i:j\in\mathcal{N}(i)} S(\tau_i^l) f_{i,j}^l \ , \tag{1}$$

subject to a set of linear constraints that ensure flow conservation [8]. In this equation $S(\tau_i^l)$ stands for the log likelihood of tracklet τ_i being the trajectory of person l and is taken to be

$$S(\tau_i^l) = \sum_{v_k\in\tau_i} \log(\varphi_k^l(t)(L+1)) \ , \tag{2}$$

where the $v_k \in \tau_i$ stand for the grid cells that compose τ_i and $\varphi_k^l(t)$ is an appearance-based probability that the person occupying grid cell v_k at time t is person l.

Note that if no appearance-based information is available for any of the vertices of tracklet τ_i, all the φ_k^l will be $1/(L+1)$ indicating that the identity could be any of the $L+1$ available ones with equal probability. In such a case, all the $S(\tau_i^l)$ will zero, thus also making all connections equally likely. By contrast, even if appearance-based information is available for only one single grid cell, it will favor one connection over the others. This formulation is therefore very good at making use of appearance information that is only available once in a while as opposed to in every consecutive frame.

Since the $f_{i,j}^l$ can only be zero or one, this amounts to solving an Integer Program. This is NP-complete, but can be relaxed into a multi-commodity network flow (MCNF) problem of polynomial complexity by making the variables real numbers between zero and one. Its solution is not guaranteed to be integral. In theory, real values that are far from either zero or one may occur. In practice this happens only when appearance information is completely lacking over several tracklets. Since this only rarely happens, we simply round off non-integer results in our experiments.

3.2 From Faces to Appearance-Based Probabilities

The solution of the Integer Program of Eq. 1 depends critically on the quality of the φ_k^l appearance-based probabilities that the identity of the person occupying grid cell v_k is l, with $0 \leq l \leq L$ where L is assumed to be known. In some cases, such as when a limited number of people are known to be present, L can be assumed to be given *a priori* and representative feature vectors, or prototypes, learned offline for each person.

Fig. 3: Some of the patches detected in the sequence of Fig. 1(b). There is much variation in lighting and head pose. The non-frontal views are shown in red on the left and the frontal ones in green on the right. Both are used in the Face Identification scenario but only the latter in the Face Re-identification one.

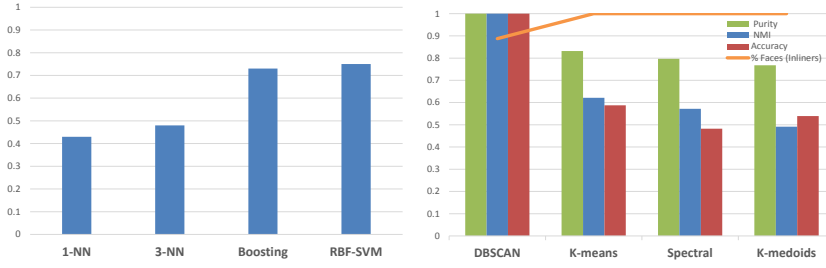
However, in more general surveillance settings, both L and the representative feature vectors must be estimated online.

In the first scenario, our run-time system estimates the desired probabilities by comparing the feature vectors it extracts from the images to the prototypes, assumed to be known prior to processing of the sequence, thus performing what we will refer to as *Face Identification*. In the second scenario, it will first create the prototypes by clustering the feature vectors and only then estimate the probabilities, an approach we will refer to as *Face Re-Identification*. In the Face Identification case, people's identities are known while in Face Re-Identification all that can be known is that different tracklets correspond to the same person. The latter is of course more challenging than the former.

In the remainder of this section, we first introduce our approach to exploiting our tracklets to extract relevant LBP features from all camera views simultaneously. We then discuss their use both for Face Identification and Re-Identification.

Extracting Relevant LBP Features Once the tracklets are created, we run a face detector in each camera view. The face detector relies on Binary Brightness Features (BBF) [30] and a cascade of strong classifiers built using a variant of AdaBoost. Instead of running the detector on the full frame, we limit its search window only to locations where a head is expected to be found.

Recall that the detections of the people are located on a discretized grid. The algorithm [29] we use to compute them relies on background subtraction for its initial processing of the images and models people as $1.75m$ tall cylinders that project to bounding boxes in the camera views. For each occupied grid cell, we only search for a face in the restricted area where the head might project to. This greatly reduces the false positive rate and allows the detector to run faster than real-time. To further eliminate false positives we enforce geometrical constraints on the faces' size given their distances from the camera. On the sequence of Fig. 1(b), this reduces the 29.15% false positive rate that standard BBF produces to a much smaller 2.5% while being 12 times



(a) (b)

Fig. 4: (a) Comparison (in terms of recognition rate) of the 4 classification methods used for face recognition on the Seq3-8p sequence: Nearest-Neighbors (1-NN and 3-NN), multi-class Boosting [33] and multi class RBF-SVM. The RBF-SVM performs slightly better than the Multi-Class Boosting. (b) Comparison of the 4 clustering methods on the fronal faces of Seq3-8p and Seq1-6p sequences: DBSCAN, K-means, Spectral, K-medoids. The DBSCAN methods performs perfectly while being able to discard outliers.

faster and allowing for real-time performance. More information about the feature extraction process can be found in [31].

For each detected face such as those of Fig. 3, we extract a vector of histograms of uniform Local Binary Pattern (LBP) features [9]. We rescale the face patches to 50×50 pixels, and extract 4×4 histograms of LBP, each of size 59. This produces a 944-dimensional floating-point vector per detected face.

Face Identification At training-time, we acquire L sequences, each one featuring a single person walking and looking at the cameras for about two minutes. We run our face-detection procedure and automatically obtain about 300 patches and extract face descriptors of his face in various orientations, which constitutes his face model.

We assign each resulting face model to one of the L identity labels. These face models constitute a train set for a multi-class RBF SVM [32] which produces an L-dimensional response vector. By combining the LBP features with the SVM classifier, the identification method becomes robust to pose variations and facial expressions. We evaluated the multi-class RBF SVM [32] against K-Nearest-Neighbors(K-NN) and multi-class boosting [33] on all the faces extracted from the Seq3-8p sequence (described in section 4.1), the results are provided in Fig. 4(a). A high threshold on the confidence score of the classifier leads to a recognition rate of 0.99, while retaining on average 36% of the detected faces.

At run-time, at each location i and time t where a face is detected, the same L-dimensional vector is computed and converted into probability $\varphi_i^l(t)$ for $0 \leq l \leq L$ using the algorithm of [34]. In the absence of a face detection, we set $\varphi_i^l(t)$ to $\frac{1}{L+1}$ for all l.

Face Re-Identification We generate the appearance groups required by the algorithm of §. 3.1 by clustering the face descriptors into L groups, where L is not provided a priori.

Clustering the whole set of detected patches yielded unsatisfactory results, due to the extended variation in head pose. To overcome this issue, we first employ a state-of-the-art face pose estimator [24] to prune the non-frontal face detections. Fig. 3 illustrates some frontal and non-frontal patches, as classified by the pose estimator, that were extracted from our test sequences.

For our purpose, we found the DBSCAN clustering algorithm [35] superior to the others we tried — K-means, K-medoids, Spectral clustering — for several reasons. The two most important ones are that it is robust to outliers and can cope with greatly uneven cluster sizes, which are recurrent issues in our experiments. In addition, the number of clusters does not need to be specified a priori, a critical requirement of our re-identification scenario. The distance measure that proved most effective in conjunction with DBSCAN is the chi-squared statistic (χ^2), which is consistent with the findings of [9]. A comparison of the four clustering methods is presented in Fig. 4(b).

4 Experiments

We use several multi-camera sequences to demonstrate that using appearance models based on facial features results in significantly more identity-preserving trajectories than either using no appearance-based information or using color-based appearance models. This is equally true both in the Identification and Re-Identification scenarios we introduced in § 3.2, that is, whether or not we have any prior knowledge of people's appearance.

In the remainder of this section, we first describe the four video sequences we use, we then introduce two baseline methods, and finally we present our comparative results.

4.1 Datasets

We present our results on four long multi-camera sequences of increasing difficulty.

- **Seq1-6p .** We first recorded a relatively simple 3500-frame sequence using 6 synchronized cameras capturing 1032×768 pixels images at 30 fps. It features six people entering a $7m \times 8m$ empty room one by one and walking around.
- **Seq2-masks .** We used the same setup to acquire a more challenging 3150-frame sequence. It features only four people but they enter, leave, and re-enter the room. Furthermore, as can be seen in Fig. 1(c), they wear surgical masks and hats, meaning that half of their faces is hidden. We ran this experiment because we are involved in a research project whose ultimate goal is to track members of a surgical team in an operating room. Since they usually all wear green scrubs, color-based features are uninformative and we wanted to check that our approach is applicable in this context.
- **Seq3-8p .** To further challenge the algorithm, we increased the number or people from four to eight and acquired a 7500-frame sequence. In the small space we are dealing with, this means that people start occluding each other very significantly in all camera views, as can be seen in Fig. 1(b).

– **Seq4-MVL [36].** The final 7840-frame sequence comes from a publicly available dataset acquired using 4 synchronized cameras. It features 5 subjects walking, entering, and leaving a furnished room. As can be seen in Fig. 1(a), the furniture often partially occludes the people and also forces them to walk close to each other. Furthermore, they all wear dark clothing and there is an abrupt change in lighting conditions in the middle of the sequence, thus making appearance-based tracking difficult.

Sequence Name	Frames	People	Tracklets	Detected Faces	Re-Identification		Identification	
					Faces	Faces / Frame	Faces	Faces / Frame
Seq1-6p	3500	6	883	744	367	0.104	285	0.08
Seq2-masks	3150	4	586	572	88	0.027	261	0.083
Seq3-8p	7500	8	2040	2379	663	0.088	790	0.105
Seq4-MVL	7840	5	10237	5981	1115	0.142	N/A	N/A

Table 1: Characteristics of the sequences we used for evaluation purposes. We also list the number of tracklets found, of detected faces, and among those of faces that are identified in both of our scenarios. The "Faces / Frame" ratio is the average number of faces identified in each frame, to be compared with the number of people actually present, listed in the third column. Note that for the Seq4-MVL sequence, there is no training data. We could therefore not test the identification scenario and the corresponding cells are marked as N/A.

The characteristics of these four sequences are summarized by Table 1. We used the sequences without any additional training data to test the Face Re-identification scenario of § 3.2 in which no appearance models are available *a priori*. We also annotated the sequences manually to generate ground-truth data and evaluate our algorithm's accuracy.

Additionally, to test the Face Identification scenario, which requires appearance models to be learned during a training phase, we acquired 30 training sequences featuring a single person walking around the capture volume and looking at the cameras once in a while. Each one of these sequences is for a different individual, including the people who actually appear in the Seq1-6p and Seq3-8p sequences. In other words, we learn $L = 30$ different models and let the algorithm decide which subset actually appears in the scene. We used almost the same procedure for the Seq2-masks sequence and learn $L = 15$ different models. The only modification is that we used only the upper parts of the faces to compute the descriptors. To this end, we built a color-based surgical-mask detector that in our experiments classified correctly 99% of the faces.

4.2 Baselines

We compare the results we obtain using our LBP-based appearance models to two baselines algorithms:

– **No Appearance.** We completely ignore appearance information and only run the Linear Programming approach of [28] to produce full trajectories, which is the first step of our processing pipeline in any event.
– **Color Identification and Re-Identification.** We replace the LBP-based appearance models by color-based appearance ones, while retaining the rest of the algorithm. By analogy and depending on whether we have *a priori* knowledge of people's appearance, we will refer to this approach as *Color Identification* or *Color Re-Identification*.

When using color, we take our feature vectors to be color histograms in CIE-LAB color space. They are obtained by averaging those computed in all views in which the person is visible. We compute distances between them in terms of the Jensen-Shannon divergence, both because it is always defined and because it returns a value between zero and one that can be converted into a probability of corresponding to the same person.

As discussed in § 3.2, the ground plane is represented by a discretized grid and our tracklets define which grid cells are occupied. Furthermore, to each occupied one in each camera view, corresponds a bounding box such as the colored rectangles overlaid on Fig. 1. We therefore build the color histogram corresponding to a person occupying a cell using the foreground pixels found by the background subtraction we use for our initial image processing [29] within the corresponding box. To increase robustness to occlusions, we use the bounding boxes corresponding to all occupied cells to compute an occlusion map, which we use to discount pixels likely to be occluded.

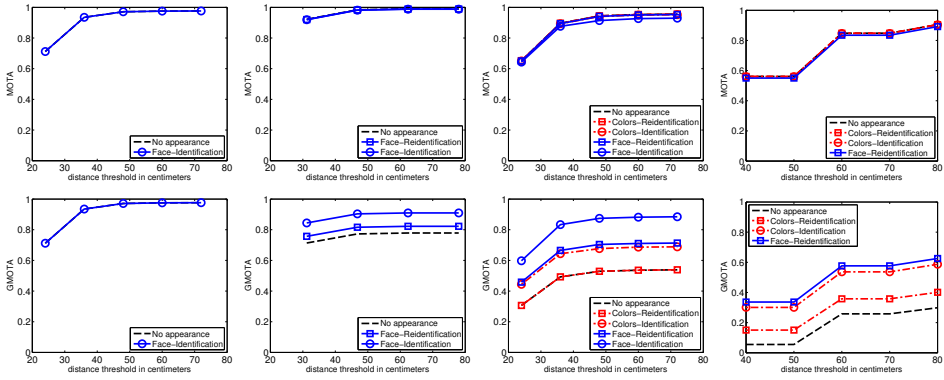
In both the Color Re-Identification and Color Identification scenarios, we use the same approach as for the LBP-vectors to either cluster the descriptors into L separate identities or to use the L predefined color templates. As opposed to the Face Identification, for colors we do not have training data. Therefore, we generated the color templates from the sequences themselves.

4.3 Results

The top of Fig. 5 depicts our results and those of our baselines in terms of the MOTA CLEAR metric [37], which is designed to evaluate performance both in terms of tracking accuracy and identity preservation. MOTA stands for *Multiple Object Tracking Accuracy* and is defined as

$$\text{MOTA} = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t} \quad , \tag{3}$$

where g_t is the number of ground truth detections, m_t the number of misdetections, fp_t the false positive count and mme_t the number of *instantaneous* identity switches. The MOTA values are plotted as functions of the ground-plane distance threshold we use to assess whether a detection corresponds to a ground-truth person. They are uniformly high for all 4 sequences because this metric is not discriminative enough. To see why, consider a case where the identities of two subjects are switched in the middle of a sequence. The MOTA score is decreased because mme_t is one instead of zero, but not by much even though the identities are wrong half of the time.



Seq1-6p Seq2-masks Seq3-8p Seq4-MVL

Fig. 5: MOTA (top) and GMOTA (bottom) scores for the four sequences described in § 4.1. Dashed black curves depict the no-appearance baseline, dashed red ones the color-based baseline, and the solid blue ones our approach. The circles denote the identification scenario and the rectangles the re-identification one, which is more challenging. Because the MOTA scores are relatively insensitive to a few identity switches, they are very similar for all scenarios. By contrast the GMOTA scores show a marked improvement when the facial descriptors are used, as discussed in § 4.3.

To remedy this, we use the more demanding GMOTA metric [8], which is defined as

$$\text{GMOTA} = 1 - \frac{\sum_t (m_t + fp_t + gmme_t)}{\sum_t g_t} \quad , \tag{4}$$

where $gmme_t$ now is the number of times in the sequence where the identity is wrong. The bottom of Fig. 5 depicts the GMOTA values for the four sequences and we discuss them individually below.

- **Seq1-6p .** Since the people enter the room one by one and do not exit, the baseline algorithm that computes tracklets while ignoring appearance cues does very well. As a result, introducing appearance does not improve the tracking results. However, by using facial cues the system can now keep track of people's identities. What this example also shows is that the LP approach we use to compute our tracklets is quite effective and therefore forms a good basis on which to build the rest of our approach.
- **Seq2-masks .** Since here, people exit and re-enter the room, appearance information is now required for identification and re-identification. Even though the people wear masks, the facial descriptors boost performance in re-identification scenario, and even more in the identification scenario as could be expected.
- **Seq3-8p .** Appearance information becomes even more important as the number of people increases and identity switches become possible when people come close to each other. As a result, the no-appearance baseline is much worse than before and the boost delivered by the facial descriptors even more significant. By comparison,

the color descriptors also provide a boost but it is much smaller. Note also that because the facial descriptors are much more powerful than color ones, the Face-Reidentification curve is higher than the Color-Identification one, even though the task is much harder.

– **Seq4-MVL .** This final sequence is more challenging than the others, in part because of the numerous occluders that hide the people. As a result, the curves are all lower but their ordering remains exactly the same as before. We do not plot the Face-Identification scores because we did not have training data we could use to learn the required prototype. However, we were able to compute the Color-Identification ones by manually extracting five color prototypes per person and, again. they are worse than the face re-identification ones.

In short, whatever the scenario chosen, using the face-based features is much more effective than using the color-based ones. And, unsurprisingly, appearance-models that have been computed off-line during a training phase boost performance.

These results are to be considered in light of how few faces actually are identified. As can be seen in the "Faces / Frame" columns of Table 1, even though there are 4 to 8 faces to be found in most temporal frames, only 0.027 to 0.142 are actually found on average. In other words, only a very small fraction of the faces are recognized but this is sufficient for our tracking algorithm.

4.4 Real-Time Implementation

We deployed a real-time version of our algorithm in one room of our laboratory. Its performance on a quad-core 3.2 GHz PC are summarized by Table. 2. The video feed is processed in 50-frame batches at a framerate of 15 Hz. In practice, this means that the result is produced with a constant $3.4s$ delay, making it completely acceptable for many broadcasting or even surveillance applications. The resolution of the images is 1032×778 pixels and we downscale them by a factor two, for people tracking while retaining the original resolution for face identification.

Algorithm	**Frames / second**
People Detection	30
Tracklets Construction	70.4
Face Detection	197.4
Face Identification	116.1
Multi-Commodity Network Flow Tracking	150

Table 2: Processing rate of individual components of our pipeline expressed in frames per second.

To achieve these run-times, we optimized the publicly available code [28,29] we use to detect people and compute the tracklets by parallelizing some parts of it and running them in separate threads.

Batch Trajectories. Our implementation uses both color and facial information, whenever available. To this end, we first we first used the KSP tracker [28] to find a set of trajectories in the current batch. We then eliminate empty grid cells from further consideration and run the Multi-Commodity Network Flow (MCNF) method on the remaining ones using color information only [8]. The required color templates are built on the fly for each tracklet connected to the source node. We allow one extra group layer for people who cannot be associated to a color template, possibly due to occlusions. This step results in a set of color-labeled trajectories.

If faces can be detected and recognized, then the MCNF method can also be used to exploit facial information as discussed above and assign trajectories to $L+1$ groups, where L is the number of uniquely identified faces in the batch. The output is a set of trajectories with potential identities that is matched to the color-labeled trajectories of the previous step. In short, each trajectory in the batch is given a color label and face identity if the necessary appearance information is available.

Joining the batches To enforce temporal consistency across batches, the trajectories extracted from consecutive overlapping batches are joined together using the Hungarian algorithm [38]. The distance between two trajectories is calculated as the sum of the Euclidean distances in the overlapping frames plus a penalty for identity and color mismatches. If no appearance information is associated to the new trajectories, the labels of the old ones are propagated to them.

5 Conclusion

We have proposed a novel and principled algorithm that combines face-recognition technology with a state-of-the-art approach to multiple-people tracking to compute identity-preserving trajectories over long periods of time. In this way, we proved that, even though faces may be detected and recognized only rarely, this is sufficient to track, identify, and re-identify individuals in a real-world scenario.

References

1. Breitenstein, M., Reichlin, F., Leibe, B., Koller-Meier, E., Van Gool, L.: Online Multi-Person Tracking-By-Detection from a Single Uncalibrated Camera. IEEE Transactions on Pattern Analysis and Machine Intelligence (2010)
2. Lu, W.L., Ting, J.A., Murphy, K.P., Little, J.J.: Identifying Players in Broadcast Sports Videos Using Conditional Random Fields. In: Conference on Computer Vision and Pattern Recognition. (2011)
3. BenShitrit, H., Berclaz, J., Fleuret, F., Fua, P.: Tracking Multiple People Under Global Apperance Constraints. In: International Conference on Computer Vision. (2011)
4. Yang, B., Nevatia, R.: Multi-target tracking by online learning of non-linear motion patterns and robust appearance models. In: Conference on Computer Vision and Pattern Recognition. (2012)
5. Zhang, W., Shan, S., Gao, W., Chen, X., Zhang, H.: Local Gabor Binary Pattern Histogram Sequence (LGBPHS): A Novel Non-Statistical Model for Face Representation and Recognition. In: International Conference on Computer Vision. (2005)
6. Lin, F.C., Denman, S., Chandran, V., Sridharan, S.: Automatic tracking, super-resolution and recognition of human faces from surveillance video. In: IAPR Conference on Machine Vision Applications. (2007)
7. Schwartz, W., Huimin, G., Jonghyunand, C., Davis, L.: Face Identification Using Large Feature Sets. IEEE Transactions on Image Processing (2012)
8. BenShitrit, H., Berclaz, J., Fleuret, F., Fua, P.: Multi-Commodity Network Flow for Tracking Multiple People. IEEE Transactions on Pattern Analysis and Machine Intelligence (2012) Submitted for publication. Available as technical report EPFL-ARTICLE-181551.
9. Ahonen, T., Hadid, A., Pietikïinen, M.: Face Description with Local Binary Patterns: Application to Face Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (2006)
10. Storms, P., Spieksma, F.: An LP-Based Algorithm for the Data Association Problem in Multitarget Tracking. Computers and Operations Research (2003)
11. Jiang, H., Fels, S., Little, J.: A Linear Programming Approach for Multiple Object Tracking. In: Conference on Computer Vision and Pattern Recognition. (2007)
12. Perera, A., Srinivas, C., Hoogs, A., Brooksby, G., Wensheng, H.: Multi-Object Tracking through Simultaneous Long Occlusions and Split-Merge Conditions. In: Conference on Computer Vision and Pattern Recognition. (2006)
13. Zhang, L., Li, Y., Nevatia, R.: Global Data Association for Multi-Object Tracking Using Network Flows. In: Conference on Computer Vision and Pattern Recognition. (2008)
14. Andriluka, M., Roth, S., Schiele, B.: Monocular 3D Pose Estimation and Tracking by Detection. In: Conference on Computer Vision and Pattern Recognition. (2010)
15. Pirsiavash, H., Ramanan, D., Fowlkes, C.: Globally-Optimal Greedy Algorithms for Tracking a Variable Number of Objects. In: Conference on Computer Vision and Pattern Recognition. (2011) Code available at http://www.ics.uci.edu/%7edramanan/.
16. Andriyenko, A., Schindler, K., Roth, S.: Discrete-Continuous Optimization for Multi-Target Tracking. In: Conference on Computer Vision and Pattern Recognition. (2012)
17. Kalal, Z., Mikolajczyk, K., Matas, J.: Face-TLD: Tracking-Learning-Detection Applied to Faces. International Conference on Image Processing (2010)
18. Cohen, A., Pavlovic, V.: An efficient ip approach to constrained multiple face tracking and recognition. In: ICCV Workshops. (2011)
19. Zhang, C., Zhang, Z.: A Survey of Recent Advances in Face Detection. Technical report, Microsoft Research (2010)

20. Zhao, W., Chellappa, R., Phillips, P., Rosenfeld, A.: Face recognition: A literature survey. ACM Computing Surveys (2003)
21. Li, S.: Handbook of face recognition. Springer (2011)
22. Viola, P., Jones, M.: Rapid Object Detection Using a Boosted Cascade of Simple Features. In: CVPR. (2001)
23. Abramson, Y., Freund, Y.: SEmi-Automatic VIsual LEarning (SEVILLE): Tutorial on Active Learning for Visual Object Recognition. In: Conference on Computer Vision and Pattern Recognition. (2005)
24. Zhu, X., Ramanan, D.: Face detection, pose estimation, and landmark localization in the wild. In: Conference on Computer Vision and Pattern Recognition. (2012)
25. Turk, M., Pentland, A.: Eigenfaces for Recognition. Journal of Cognitive Neuroscience **3** (1991) 71–86
26. Belhumeur, P., Hespanha, J., Kriegman, D.: Eigenfaces Vs. Fisherfaces: Recognition Using Class Specific Linear Projection. IEEE Transactions on Pattern Analysis and Machine Intelligence **19** (1997) 711–720
27. Chan, C., Tahir, M., Kittler, J., Pietikainen, M.: Multiscale local phase quantization for robust component-based face recognition using kernel fusion of multiple descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence (2013)
28. Berclaz, J., Fleuret, F., Türetken, E., Fua, P.: Multiple Object Tracking Using K-Shortest Paths Optimization. IEEE Transactions on Pattern Analysis and Machine Intelligence (2011) Code available at http://cvlab.epfl.ch/software/ksp.
29. Fleuret, F., Berclaz, J., Lengagne, R., Fua, P.: Multi-Camera People Tracking with a Probabilistic Occupancy Map. IEEE Transactions on Pattern Analysis and Machine Intelligence (2008) Code available at http://cvlab.epfl.ch/software/pom.
30. Abramson, Y., Steux, B., Ghorayeb, H.: YEF Real-Time Object Detection. In: International Workshop on Automatic Learning and Real-Time (ALaRT). (2005)
31. Zervos, M.: Multi-Camera Face Detection and Recognition Applied to People Tracking. Master's thesis, Ecole Polytechnique Fédérale de Lausanne (2013)
32. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology (2011) Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.
33. Saberian, M.J., Vasconcelos, N.: Multiclass Boosting: Theory and Algorithms. In: Advances in Neural Information Processing Systems. (2011)
34. Wu, T., Lin, C., Weng, R.C.: Probability Estimates for Multi-Class Classification by Pairwise Coupling. Journal of Machine Learning Research (2004)
35. Ester, M., Kriegel, H., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Knowledge Discovery and Data Mining. (1996)
36. Mandeljc, R., Kristan, S.K.M., Per, J.: Tracking by Identification Using Computer Vision and Radio. Sensors (2012) Dataset available at http://vision.fe.uni-lj.si/research/mvl_lab5/.
37. Bernardin, K., Stiefelhagen, R.: Evaluating Multiple Object Tracking Performance: the Clear Mot Metrics. EURASIP Journal on Image and Video Processing (2008)
38. Kuhn, H.: The Hungarian Method for the Assignment Problem. Naval Research Logistics Quarterly (1955)