Pattern Recognition 64 (2017) 361–373

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

A Coupled Hidden Markov Random Field model for simultaneous face clustering and tracking in videos

Baoyuan Wu[a,b,*], Bao-Gang Hu[a], Qiang Ji[c]

[a] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, PR China
[b] Visual Computing Center of King Abdullah University of Science and Technology, 23955-6900, Saudi Arabia
[c] Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180, USA

ARTICLE INFO

Keywords:
Face clustering
Face tracking
Coupled Hidden Markov Random Field

ABSTRACT

Face clustering and face tracking are two areas of active research in automatic facial video processing. They, however, have long been studied separately, despite the inherent link between them. In this paper, we propose to perform simultaneous face clustering and face tracking from real world videos. The motivation for the proposed research is that face clustering and face tracking can provide useful information and constraints to each other, thus can bootstrap and improve the performances of each other. To this end, we introduce a Coupled Hidden Markov Random Field (CHMRF) to simultaneously model face clustering, face tracking, and their interactions. We provide an effective algorithm based on constrained clustering and optimal tracking for the joint optimization of cluster labels and face tracking. We demonstrate significant improvements over state-of-the-art results in face clustering and tracking on several videos.

1. Introduction

Facial images provide vital identification information in applications of automatic video analysis [1]. As the basis of reliable face recognition in videos are two critical steps, namely *face clustering*, where face images are partitioned into different clusters, and *face tracking*, where sequences of face images are associated. However, reliably clustering and tracking of faces in unconstrained videos is a challenging problem, which is complicated by drastic variations in pose, illuminations, view points, camera movements and occlusions that frequently occur in actual videos.

Most previous works treat face clustering and tracking in videos as two individual problems. Examples of recent works on face clustering in videos include [2–4], and those of recent works on face tracking in videos include [5,6]. Yet, the two problems are intimately related with each other and can provide useful information and constraints to each other, thus can bootstrap and improve the performances of each other. Fig. 1 exemplifies the benefits of simultaneous face clustering and tracklet linking[1] in the first case (left), incorrect clustering of faces (in this case, separation of faces of the same person into two clusters 1 and 2) can be avoided with the knowledge that there is a high likelihood that they are in the same long track. In the second case (right), linking tracklets without considering their cluster labels leads to incorrect association of tracklets from clusters 1 and 2 together. Thus it is advantageous to solve face clustering and tracklet linking together.

In this work, we address the problem of *simultaneous* clustering and linking of tracklets in videos. We introduce a Coupled Hidden Markov Random Field (CHMRF) model by coupling two Hidden Markov Random Field (HMRF) models [7]. These two models formulate the clustering and tracklet linking respectively and the links between them capture their interactions. Given CHMRF, we formulate this joint problem as a Bayesian inference problem, and provide an efficient coordinate-descent solution. Fig. 2 provides an overview of the proposed method and its major steps. Specifically, from the detected face tracklets with similar appearances and adjacent spatial locations in consecutive frames, our method iterates between two steps. (1) *Face clustering*: Recovering face cluster labels using constrained clustering with constraints from both the intermediate tracklet linking results and the spatiotemporal knowledge in videos; (2) *Face tracklet linking*: Finding long tracks of faces by linking face tracklets that are consistent in motion, appearance, and with the intermediate tracklet clustering results.

The resulting longer tracks of distinct faces sharing same cluster labels constitute the basis for multi-face tracking and identity maintenance, which are important tasks in video indexing, retrieval and summarization. The contributions of this work are thus highlighted in

* Corresponding author at: National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, PR China.
E-mail addresses: wubaoyuan1987@gmail.com (B. Wu), bghu@nlpr.ia.ac.cn (B.-G. Hu), qji@ecse.rpi.edu (Q. Ji).

[1] *Face tracklet* indicates the short sequence of detected faces. In this work we consider *tracklet linking*, which is a branch of *tracking*. So it should be noted that the words *tracklet linking* and *tracking* have different meanings hereafter.

http://dx.doi.org/10.1016/j.patcog.2016.10.022
Received 24 April 2016; Received in revised form 21 September 2016; Accepted 19 October 2016
Available online 26 October 2016

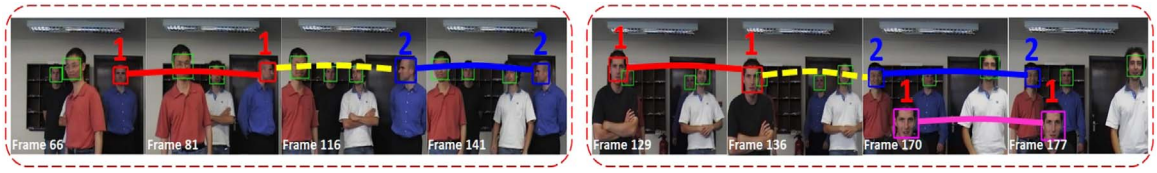

Fig. 1. The benefit of simultaneous face clustering and tracklet linking. Detected face tracklets are indicated by bounding boxes connected with solid lines (we only highlight a few detected tracklets for the sake of presentation). Linkings and cluster labels of tracklets are indicated by the dashed curves and numbers over the bounding boxes, respectively. (Left) Without considering tracklet linking, tracklets in the same track are incorrectly partitioned into different clusters (the frames are extracted from the Turning video [5]). (Right) Without considering clustering labels, tracklets of different clusters are linked incorrectly (the frames are extracted from the Frontal video [5]).

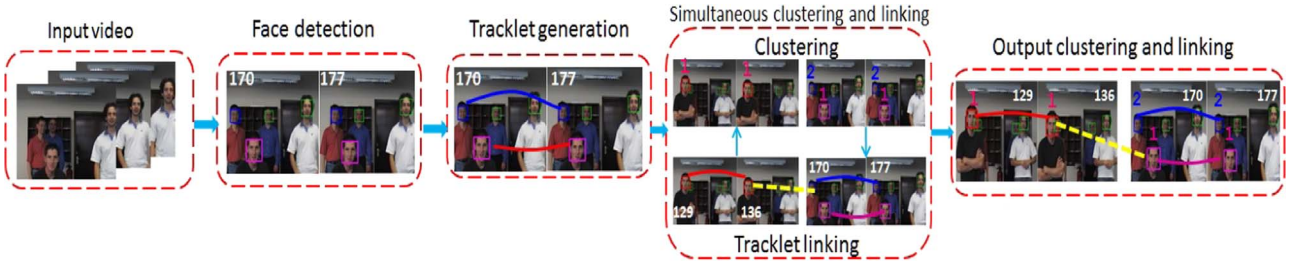

Fig. 2. Overall workflow of our method for simultaneous face clustering and tracklet linking. With an input video (here we use the Frontal video [5]), we first detect all faces in each frame, then form face tracklets from adjacent frames. The face tracklets are iteratively clustered and linked into longer tracks in a bootstrapping manner, with the final output of the algorithm being the complete long face tracks with cluster labels.

three aspects: (a) we present a systematic approach to jointly solving two related tasks, i.e., face clustering and face tracklet linking, by taking advantage of both the prior knowledge extracted from videos and the additional information from each task to boost their overall performances; (b) we introduce a coupled hidden Markov random field model, and develop algorithms to perform efficient learning and inference; (c) we demonstrate that the proposed method outperforms state-of-the-art face clustering and tracklet linking methods on realistic benchmark face videos.

The rest of the paper is organized as follows. After reviewing related works in Section 2, we describe the CHMRF model in Section 3, and present the optimization in Section 4. The neighborhood system is demonstrated in Section 5. Experimental evaluations and comparisons of our method are reported in Section 6 and Section 7 concludes the paper with discussion and future work.[2]

2. Related works

Face clustering and face tracking are two important problems in video processing. In the following, we review the existing works on these two problems independently, then introduce attempts to combine them together.

Face clustering in videos has been explored in many previous works. They can be grouped into two categories: purely data-driven methods and clustering with prior knowledge. Most data-driven methods are fully unsupervised, and focus on obtaining a good distance metric [9–13]. Fitzgibbon and Zisserman [9] proposed an affine invariant distance measure to achieve robustness to face pose changing. They later [10] extended their work to a Joint Manifold Distance (JMD), where each subspace represents a set of facial images of the same person. Wang et al. [11] proposed a Manifold-Manifold Distance (MMD), in which a nonlinear manifold is divided into several local linear subspaces. MMD integrates the distances between pair of subspaces respectively from one of the involved manifolds. Hu et al. [12] introduced a between-set distance called Sparse Approximated Nearest Point (SANP) distance, where the dissimilarity of two sets is measured as the distance between their nearest points. Arandjelovic and Cipolla [13] clustered faces over face appearance manifolds in an anisotropic manifold space which exploits the coherence of dissimilarities between manifolds. In addition to fully unsupervised methods, another kind of data-driven methods tries to utilize some partial supervision to help clustering. Prince and Elder [14] combined clustering with a Bayesian approach to count the number of different people that appear in a collection of face images. A generative model describing the face manifold is learned from the training data. Du and Chellappa [15] presented an on-line context-aided face association method, where multiple contextual features are embedded into a conditional random field (CRF) model. Wolf et al. [16] described a set-to-set similarity measure named matched background similarity (MBGS). It can tell the differences between images with similar background. Such that it is robust to the changes on pose, lighting, and viewing conditions.

The main drawback of data-driven methods is the expected unstable performance due to the drastic variations of faces in real videos. To alleviate this, prior knowledge could be exploited to guide the clustering to achieve robustness and increase the generalization ability. Berg et al. [17] considered using extra information to enhance face clustering, where the faces are collected from web news pages. A set of names automatically captured from associated news captions are employed to supervise the clustering. However, such text-based labels are not always available for faces in videos. Fortunately, there is readily useful prior knowledge for face clustering in videos: *faces in the same tracklet must belong to the same person* (*must-link*), *no matter how different their appearances look like; on the other hand, if two tracklets overlap in some frames, then faces from them must be from different persons* (*cannot-link*), *no matter how similar they look like*. Thus, we can easily obtain many must-link and cannot-link constraints from face tracklets without much extra cost. However, few works in face clustering have exploited such constraints. Vretos et al. [3] exploited such constraints to modify the distance matrix. However, the method is very computationally expensive. Cinbis et al. [2] also proposed a metric learning method, called unsupervised logistic discriminative metric learning (ULDML). However, its metric learning is independent of the clustering process, which may lead to non-robust clustering performance. Xiao et al. [18] utilized the pairwise constraints to learn a low rank representation for facial images. Furthermore, Cao et al. [19,20] considered the face clustering in multi-view learning, where the pairwise constraints are used in both representation learning and clustering. In our previous work, i.e., HMRF-pc [4], the initial constraints are propagated based on con-

[2] Preliminary versions of this work have been published in [4,8].

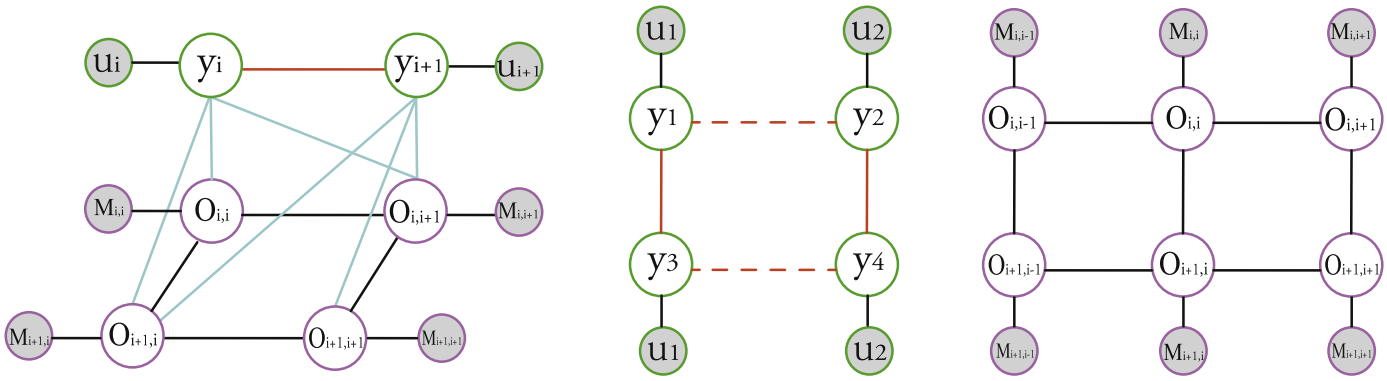

Fig. 3. A graphical illustration of the proposed models. Transparent nodes indicate hidden variables, while shaded nodes represent observed variables; green nodes denote variables for clustering, while purple nodes represent variables for tracklet linking. (left) The CHMRF model consists of two HMRF models, which are connected by cyan lines to embed the dependencies between clustering and tracklet linking variables; (middle) the HMRF model for clustering, with the solid red lines being positive correlations, while dashed red line being negative correlations; (right) the HMRF model of tracklet linking. Note that the nodes in each row/column of O are fully connected, but here we ignore some lines in (left) and (right) for clarity. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

straint-level smoothness. Then a HMRF based clustering model is proposed to incorporate the cluster assumption and constraint satisfaction together. It can be considered as the clustering part of the proposed unified model CHMRF (see Fig. 3). However, like other existing methods, HMRF-pc also performs face clustering individually, without the constraints from the results of face tracklet linking. In the literature of constrained clustering, many methods have been proposed to exploit pairwise constraints to guide clustering, such as constrained K-means (COP-KMEANS) [21], constrained complete-link (CCL) [22], metric learning with side information [23], constrained EM [24] and penalized probabilistic clustering (PPC) [25]. The most related method to the clustering part of CHMRF is HMRF-KMeans [26]. It also adopted the HMRF model. The main difference between them is how to utilize the initial constraints. HMRF-KMeans propagates the influence of initial constraints through metric learning, assuming that the cluster follows certain distribution such as Gaussian, or multinomial distribution. If the true distribution is far from the assumed distribution, the learned metric is not expected to help or may even harm the clustering. In contrast, CHMRF explicitly propagates initial constraints based on two distribution-independent assumptions, including constraint consistency and smoothness. Besides, CHMRF exploits the constraints from tracklet linking results, while such constraints are ignored in HMRF-KMeans.

Multi-face tracking is another important task in facial video processing. Techniques for face tracking can be divided into monolithic tracking and tracklet-based tracking. Monolithic tracking performs target tracking continuously until the target is lost; tracklet-based tracking, on the other hand, performs target tracking by linking the tracklets (resulted from monolithic tracking) to form longer track. The commonly used monolithic multiple hypothesis tracking (MHT) methods include Kalman filtering and particle filtering. These methods work well under constrained or fixed environment and they could lead to broken tracks when the targets or the environments undergo significant changes. As a result, recently there is an increasing interest in tracklet based tracking (e.g., [27–30]), which links the shorter tracklets by firstly constructing pairwise similarities between tracklets (based on appearance as in [27,28], motion smoothness [29] or entry/exit maps [29]), followed by optimally linking (using the Hungarian algorithm [31] for optimal matching of bi-partite graph [30,32] or as a Bayesian inference [28]). Compared to the monolithic tracking solutions, the tracklet based methods are more robust and suitable for tracking multiple objects in heavily occluded scenes for a long period of time. The tracklet-based methodology has been employed for face tracking in [5,6]. The PHD-MT [5] firstly adopted a probability hypothesis density (PHD) filter to compensate miss-detections and to remove noises, then the filtered faces are associated using graph matching. However, the method was designed for a fixed scene. Roth et al. [6] developed a multi-stage tracklet linking method, where different cues, including face id, classifier and constraint cues, are adopted to help linking. Some tracklet-based methods for tracking other general objects are also developed. Singh et al. [33] exploited both the tracklets with high detection confidence and the unassociated detections with low confidence to improve the performance of tracklet linking. Huang et al. [32] presented a hierarchical association framework with three levels by using different features. However, this framework was only suitable to the scenario of single camera. Song et al. [34] proposed a stochastic graph evolution framework to utilize the statistics of the tracklets. Li et al. [35] formulated tracklet linking as a joint learning problem of ranking and classification. A boosting algorithm called HybridBoost was proposed to learn the tracklet affinities, such that the relative weights of different features (appearance, location, frame gap) can be automatically learned. However, the supervised learning also limited HybridBoost to be more suitable to the tracking with fixed camera. Wang et al. [36] proposed to learn a metric for each tracklet, which is used to refine the initial tracklet and to compute the affinity between tracklets. Zhang et al. [37] utilized the group state (whether an object belongs to a group) to detect the merging or splitting event, which is further used to help the tracklet fusion across different cameras. Bae and Yoon [38] considered the tracklet confidence to help linking.

Although there have been a large amount of works in the separate literatures of face clustering and tracklet linking, few works have tried to exploit their interdependencies to improve their performances. Two pioneering works were proposed in [30,39]. However, the referred clustering is motion clustering rather than face clustering, and serves as an assistance for tracklet linking. Most importantly, both methods were developed for the simplified context with fixed camera. To our best knowledge, no previous work has been explored in a more challenging scenario with many camera motions and occlusions for simultaneous face clustering and tracklet linking.

3. A unified graphical model of simultaneous clustering and tracklet linking

3.1. Problem formulation

We assume that a long video has been pre-processed to obtain a set of m face tracklets $U = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m)$. Each tracklet $\mathbf{u}_i$ is represented as a list of triples collected from a sequence of n_i continuous video frames, as $\mathbf{u}_i = \{t_j^{(i)}, \mathbf{x}_j^{(i)}, \mathbf{l}_j^{(i)}\}_{j=1}^{n_i}$, where $t_j^{(i)}$ is the frame index, $\mathbf{x}_j^{(i)}$ is the corresponding appearance feature, and $\mathbf{l}_j^{(i)}$ represents the location and scale of the bounding box of the detected face in this tracklet, respectively. Furthermore, we use $\mathbf{t}^{(i)}$, $X^{(i)}$ and $L^{(i)}$ to represent the

ensemble of $t_j^{(i)}$, $\mathbf{x}_j^{(i)}$ and $\mathbf{l}_j^{(i)}$ of tracklet $\mathbf{u}_i$, respectively. The similarities between every pair of tracklets (details in Section 4.2) are also computed and saved in an $m \times m$ matrix M.

Our goal is to *simultaneously* partition the facial images into distinct clusters and link the tracklet into longer tracks, based on cues from face appearances and motion trajectories. For simplicity, we assume that the total cluster number of K is known a priori.[3] We denote the cluster labels of the tracklets as a vector $\mathbf{y} = (y_1, y_2, \ldots, y_m)$ with each $y_i \in \{1, 2, \ldots, K\}$. The linking relations of tracklets are represented with a matrix $O \in \{0, 1\}^{m \times m}$, where $O_{ij} = 1$ if and only if tracklets $\mathbf{u}_i$ and $\mathbf{u}_j$ are adjacent in a track with $\mathbf{u}_i$ precedes $\mathbf{u}_j$, and $O_{ii} = 1$ if and only if tracklet $\mathbf{u}_i$ is the last tracklet in a long track. All major notations used in this work are summarized in Table 1.

The probabilistic inter-dependencies between U, M, $\mathbf{y}$ and O are modeled with a Coupled Hidden Markov Random Field (CHMRF), where U and M are observable variables, while $\mathbf{y}$ and O are latent variables. Their joint probability distribution, along with model parameters $\Lambda = (\{\mu_i\}_{i=1}^K, \{\Sigma_i\}_{i=1}^K, \beta)$ (specified later), are defined as:

$$P(U, M, \mathbf{y}, O|\Lambda) = P(U|\mathbf{y}; \Lambda)P(M|O)P(\mathbf{y}, O|\Lambda), \tag{1}$$

where $P(U|\mathbf{y}; \Lambda)$ and $P(M|O)$ are two unary potential functions, which will be introduced in detail in Section 3.2; $P(\mathbf{y}, O|\Lambda)$ is referred to as the pairwise potential function, which will be specified in Section 3.3.

A graphical illustration of CHMRF and its variants are shown in Fig. 3. In Fig. 3-left, both cluster labels $\mathbf{y}$ and linkings O are unobserved, and the lines between them denote their dependencies: if $O_{ij} = 1$, then y_i should equal to y_j; if $y_i \neq y_j$, then O_{ij} should be 0. If O is observed, then the dependencies between $\mathbf{y}$ and O are transformed to the pairwise constraints between $\mathbf{y}$, and the unified model will reduce to the pure clustering model briefly illustrated in Fig. 3-middle. If $\mathbf{y}$ is observed, then the dependencies between $\mathbf{y}$ and O are transformed to the pairwise constraints between O, and the unified model will reduce to the pure tracklet linking model briefly illustrated in Fig. 3-right. The unified model corresponds to the simultaneous clustering and linking problem, and the second model represents the separate clustering problem, while the last one denotes the separate linking problem. More details will be presented in Section 4.

It has to be clarified that we have used a new name CHMRF to describe the joint model, while we adopted the name HMRF in our previous conference paper [8]. There are two reasons of the name change. Firstly, obviously the structure of the proposed joint model CHMRF satisfies the definition of HMRF. However, in conventional HMRF, all hidden nodes are homogeneous, i.e., they indicate same meanings. But it is not the case in CHMRF. There are two types of hidden nodes, including the clustering label nodes and the tracklet linking nodes. And the neighborhood systems within themselves, as well as the one between them, are different. Moreover, in conventional HMRF, all label nodes can be updated simultaneously, while in our model two types of label nodes are updated sequentially. To highlight the differences on both hidden nodes and optimization with the conventional HMRF, in this manuscript we decide to use the new name CHMRF, which combines two heterogeneous HMRFs to a unified model.

3.2. Unary potential functions

Following Eq. (1), we define the two unary potential functions as follows. Firstly, we model the conditional distribution of the appearances of facial images in the tracklets given their cluster labels with Gaussian distribution,

[3] For example, for videos from TV episodes, K can be determined using the number of major characters obtained in the cast.

Table 1
Main notations used in this paper.

Symbols	Descriptions
$U = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m)$	A set of m face tracklets
$\mathbf{u}_i = \{t_j^{(i)}, \mathbf{x}_j^{(i)}, \mathbf{l}_j^{(i)}\}_{j=1}^{n_i}$	One tracklet of n_i faces
$\mathbf{t}^{(i)} = (t_1^{(i)}, t_2^{(i)}, \ldots, t_{n_i}^{(i)})'$	Frame indexes of the bounding boxes for the ith tracklet
$X^{(i)} = (\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, \ldots, \mathbf{x}_{n_i}^{(i)})$	The appearance of the bounding boxes for ith tracklet
$L^{(i)} = (\mathbf{l}_1^{(i)}, \mathbf{l}_2^{(i)}, \ldots, \mathbf{l}_{n_i}^{(i)})$	Locations and scales of the bounding boxes for the ith tracklet
$\mathbf{y} = (y_1, y_2, \ldots, y_m)$	The cluster labels of all m tracklets
$y_i \in \{1, 2, \ldots, K\}$	K is the number of clusters
$V \in \mathrm{R}^{m \times m}$	Pairwise constraints among the cluster labels of tracklets
$O \in \{0, 1\}^{m \times m}$	The linking matrix
$W \in \mathrm{R}^{n \times n}$	Pairwise constraints among the cluster labels of all detected faces
$M = \{M_{ij}\} \in \mathrm{R}^{m \times m}$	The observation matrix for O
$M_{ij} = f(\mathbf{u}_i, \mathbf{u}_j)$	The similarity between $\mathbf{u}_i$ and $\mathbf{u}_j$

$$P(U|\mathbf{y}; \Lambda) = \prod_{i=1}^{m} \prod_{j=1}^{n_i} \mathcal{N}(\mathbf{x}_j^{(i)}|\mu_{y_i}, \Sigma_{y_i}), \tag{2}$$

where parameters μ_{y_i} and Σ_{y_i} correspond to the cluster-specific means and covariance matrices respectively and are estimated during the optimization of $\mathbf{y}$ (described in Section 4.1.1). The second unary potential, $P(M|O)$, which captures the relation between tracklet similarities and the linking relations in long tracks, is defined as:

$$P(M|O) = \frac{1}{Z_1} \prod_{i=1}^{m} \prod_{j=1}^{m} \exp(\lambda_1 O_{ij} M_{ij}), \tag{3}$$

where O_{ij} indicates the linking relation between $\mathbf{u}_i$ and $\mathbf{u}_j$, while M_{ij} denotes the corresponding similarity. The exponent of their product measures the probability of the linking relation. $Z_1 = \sum_M \prod_{i=1}^{m} \prod_{j=1}^{m} \exp(\lambda_1 O_{ij} M_{ij})$ being the partition function. The tuning of parameter λ_1 will be described in Section 4.2.

3.3. Pairwise potential functions

The pairwise potential function captures the dependencies among the latent variables. Three types of correlations are considered in CHMRF, including correlations: (a) among tracklet linkings O; (b) among the cluster labels $\mathbf{y}$; (c) between $\mathbf{y}$ and O. So $P(\mathbf{y}, O|\Lambda)$ is defined as:

$$P(\mathbf{y}, O|\Lambda) = \frac{1}{Z_2} \prod_{i=1}^{m} \left(\psi_{p1}\left(\sum_{j=1}^{m} O_{ij}, \sum_{j \neq i} O_{ji} \right) \prod_{j \neq i} [\psi_{p2}(y_i, y_j) \psi_{p3}(y_i, y_j, O_{ij})] \right)^{\beta}, \tag{4}$$

with Z_2 being the partition function. The model parameter β controls the trade-off between the unary and the pairwise potential functions. It will be learned automatically during the optimization (described in Section 4.1).

The first pairwise potential function ψ_{p1} captures dependencies among components of O and is formulated as

$$\psi_{p1}\left(\sum_{j=1}^{m} O_{ij}, \sum_{j \neq i}^{m} O_{ji} \right) = \mathbb{I}\left(\sum_{j=1}^{m} O_{ij} = 1 \right) \mathbb{I}\left(\sum_{j \neq i}^{m} O_{ji} < 1 \right), \tag{5}$$

where $\mathbb{I}(\cdot)$ is the indicator function, whose value is 1 if the argument is true, and 0 otherwise. ψ_{p1} incorporates the high-order dependency, and is reflected by Fig. 3-right, where nodes in the same row/column are fully connected.

The second pairwise potential function ψ_{p2} models the dependencies among the cluster labels $\mathbf{y}$ and is defined as

$$\psi_{p2}(y_i, y_j) = e^{\left[\mathbb{I}(V_{ij}\geq 0)V_{ij}(\mathbb{I}(y_i=y_j)-1)+\mathbb{I}(V_{ij}<0)V_{ij}\mathbb{I}(y_i=y_j)\right]}. \quad (6)$$

The pre-computed matrix $V = \{V_{ij} | i, j = 1, \ldots, m\} \in \mathbb{R}^{m\times m}$ saves the dependencies among $\mathbf{y}$: $V_{ij} > 0$ indicates positive correlation between y_i and y_j, i.e., they are more likely to take the same value; $V_{ij} < 0$ indicates negative correlation between y_i and y_j, i.e., they are more likely to take different values; $V_{ij} = 0$ represents the cluster labels being independent. V can be seen as the neighborhood system of $\mathbf{y}$, as reflected in Fig. 3-middle. More details about the computation of V will be described in Section 5.

The last potential ψ_{p3} incorporates the dependencies between $\mathbf{y}$ and O,

$$\psi_{p3}(y_i, y_j, O_{ij}) = \exp\left(\lambda_2 O_{ij}(\mathbb{I}(y_i = y_j) - 1)\right). \quad (7)$$

It corresponds to the following constraint between $\mathbf{y}$ and O: *if* $y_i \neq y_j$, *then the configuration* $O_{ij} = 1$ *will be discouraged; if* $O_{ij} = 1$, *then* y_i=y_j *will be encouraged*. The parameter λ_2 controls influence degree between $\mathbf{y}$ and O. It can be seen as the weight of the lines which link the top layer $\mathbf{y}$ and the bottom layer O in Fig. 3-left. Its tuning will be discussed in Section 4.2.

4. Optimization

With the CHMRF formulation, our task of simultaneous clustering and tracklet linking of facial images can be formulated as maximizing the joint log likelihood,

$$\max_{\mathbf{y},O,\Lambda} \log P(\mathbf{y}, O, U, M; \Lambda) = \log P(U|\mathbf{y}; \Lambda) + \log P(M|O) + \log P(\mathbf{y}, O|\Lambda). \quad (8)$$

It can be solved by coordinate-descent method, as summarized in Algorithm 1.

Algorithm 1. Overall algorithm for simultaneous face clustering and tracklet linking.

Input: tracklets U, their similarity M, cluster number K
Output: cluster labels $\mathbf{y}$ and tracklet linking relation O
1: Initialize O based on M, using Hungarian algorithm;
2: **while** not converge **do**
3: optimize $\mathbf{y}$ and Λ with fixed O (Section 4.1);
4: optimize O with fixed $\mathbf{y}$ (Section 4.2);
5: **end while**
6: **return** $\mathbf{y}^\star$ and $O^\star$

4.1. Constrained clustering: optimizing $(\mathbf{y}, \Lambda)$ given O

This sub-problem is maximizing the complete log likelihood, as follows:

$$(\mathbf{y}^\star, \Lambda^\star) = \arg\max_{\mathbf{y},\Lambda} \log P(U|\mathbf{y}; \Lambda) + \log P(\mathbf{y}, O|\Lambda) \equiv \arg\max_{\mathbf{y},\Lambda} \log P(\mathbf{y}, O|U; \Lambda) + \log P(U|\Lambda). \quad (9)$$

It can be solved through two consecutive steps:

1. $\Lambda^\star$ is estimated by solving $\arg\max_\Lambda \log P(U|\Lambda)$ using EM algorithm;
2. given $\Lambda^\star$, compute $\mathbf{y}^\star$ by $\mathbf{y}^\star = \arg\max_{\mathbf{y}|\Lambda} \log P(\mathbf{y}, O|U; \Lambda)$.

Due to the dependencies among $\mathbf{y}$, it is difficult to solve these two problems. We resort to the *simulated field algorithm* [40], whose main idea is to decouple $\mathbf{y}$ by introducing the temporary label configurations $\bar{\mathbf{y}}$, as shown in Algorithm 2.

Moreover, before presenting more details about this algorithm, for clarity we should make some changes in the notations and formulations in this section. As we perform clustering on each face, rather than on each tracklet (as demonstrated in Section 5), the sizes of the corresponding notations should be changed, including the appearance feature matrix X, the cluster label vector $\mathbf{y}$, the neighborhood system matrix W and their involved formulations. Specifically, $X = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \mathbb{R}^{d\times n}$ include the feature vectors of n faces, and $\mathbf{y} = (y_1, \ldots, y_n)$ are the corresponding cluster labels of faces. The model formulation becomes

$$P(X, \mathbf{y}|\Lambda) = \frac{1}{Z_X(\Lambda)} \prod_{i=1}^{n} [\psi_u(\mathbf{x}_i, y_i; \Lambda)\psi_p(y_i, \mathbf{y}_{N_i}; \Lambda)], \quad (10)$$

$$\psi_p(y_i, \mathbf{y}_{N_i}|\Lambda) = \prod_{j\in N_i} e^{\left(\beta\left[\mathbb{I}(W_{ij}>0)W_{ij}(\mathbb{I}(y_i=y_j)-1)+\mathbb{I}(W_{ij}\leq 0)W_{ij}\mathbb{I}(y_i=y_j)\right]\right)}, \quad (11)$$

$$\psi_u(\mathbf{x}_i, y_i|\Lambda) = \mathcal{N}(\mathbf{x}_i|\mu_{y_i}, \Sigma_{y_i}), \quad (12)$$

where ψ_u denotes the unary potential, while ψ_p indicates the pairwise potential. $W \in \mathbb{R}^{n\times n}$ represents the neighborhood system, as explored in Section 5. N_i denotes the neighborhood set of y_i: if $W(i, j) \neq 0$, then $j \in N_i$.

Note that the fixed linking results O have been embedded into W, so in the remaining part of this subsection we ignore the notation O for clarity. In the following, we firstly present the details of simulated field algorithm in Section 4.1.1; then an efficient clustering framework is proposed to reduce the computational cost of clustering on each face, as shown in Section 4.1.2.

4.1.1. Simulated field algorithm

The main idea of simulated field algorithm [40] is: when treating a particular latent variable y_i, the states of its neighbors are fixed; as a result, the overall computation reduces to deal with independent variables. Note that simulated field algorithm is also called "mean field-like approximation" [40]. However, the difference between simulated field and mean field method is: in simulated field, for each node of interest, the influence from other nodes is fixed and computed based on constants; in mean field, the state of each node is updated based on the mean value of other nodes, then the influence from other nodes are varied. More details about their difference can be found in [40]. Specifically, simulated field algorithm consists of two parts.

1. Learning of Λ consists of two iterative steps:
   (a) simulate $\bar{\mathbf{y}}$ according to $P(\mathbf{y}|X, \bar{\mathbf{y}}^{\text{old}}; \Lambda^{\text{old}})$;
   (b) given $\bar{\mathbf{y}}$, run EM algorithm to update $P(\mathbf{y}|X, \bar{\mathbf{y}}; \Lambda)$ and Λ.
2. Based on Λ^* and $\bar{\mathbf{y}}$, $\mathbf{y}$ is approximated from $\arg\max_{\mathbf{y}} P(\mathbf{y}|X, \bar{\mathbf{y}}; \Lambda^*)$.

The main procedure is summarized in Algorithm 2.

Algorithm 2. Simulated field algorithm for HMRF based constrained clustering.

Input: data X, neighborhood system W, cluster number K, maximal iterations t_{max}
Output Λ^* and $\mathbf{y}^*$
1: Initialize $\Lambda^{(0)}$ and $\bar{\mathbf{y}}^{(0)}$ by K-means
2: **for** t=1 **to** t_{max} **do**
3: $\bar{\mathbf{y}}^{(t)} \sim P(\mathbf{y}|\bar{\mathbf{y}}^{(t-1)}, X; \Lambda^{(t-1)})$, see Section 4.1.1
4: learn $P(\mathbf{y}|\bar{\mathbf{y}}^{(t)}, X; \Lambda^{(t-1)})$ and $\Lambda^{(t)}$, see Section 4.1.1
5: **if** convergence **then**
6: **break**
7: **end if**
8: **end for**
9: $\Lambda^* = \Lambda^{(t)}$, $\mathbf{y}^* = \arg\max_{\mathbf{y}} P(\mathbf{y}|\bar{\mathbf{y}}^{(t)}, X; \Lambda^*)$.

Given $P(\mathbf{y}|X, \bar{\mathbf{y}}^{\text{old}}; \Lambda)$, simulate $\bar{\mathbf{y}}$

In the t-th iteration, $\bar{\mathbf{y}}$ are simulated from $P(\mathbf{y}|\bar{\mathbf{y}}^{(t-1)}, X; \Lambda^{(t-1)})$ in a sequential form, following the Gibbs sampling [41]. Specifically, $\bar{y}_i \sim P(y_i|\bar{\mathbf{y}}_{N_i}^{(t,i)}, X; \Lambda^{(t-1)})$ with $\bar{\mathbf{y}}_{N_i}^{(t,i)} = (\bar{y}_1^{(t)}, \ldots, \bar{y}_{i-1}^{(t)}, \bar{y}_{i+1}^{(t-1)}, \bar{y}_n^{(t-1)}) \cap N_i$.

Given $\bar{\mathbf{y}}$, learn $P(\mathbf{y}|X, \bar{\mathbf{y}}; \Lambda)$ and Λ.

Given $\bar{\mathbf{y}}^{(t)}$, the couplings among $\mathbf{y}$ are decomposed, then $P(\mathbf{y}|X, \bar{\mathbf{y}}; \Lambda)$ and Λ can be iteratively updated by EM algorithm. As per the suggestion in [40], only one iteration of EM is conducted in this part.

E step: inference of $P(\mathbf{y}|X, \bar{\mathbf{y}}; \Lambda)$, *given* $\bar{\mathbf{y}}^{(t)}$ *and* $\Lambda^{(t-1)}$

Given $\bar{\mathbf{y}}$, $P(\mathbf{y}|\bar{\mathbf{y}}^{(t)}; \Lambda^{(t-1)}) = \prod_{i=1}^{n} P(y_i|\bar{\mathbf{y}}_{N_i}^{(t)}, \mathbf{x}_i; \Lambda^{(t-1)})$. As a result, the posterior probability over each face can be computed independently, as follows:

$$P(y_i|\mathbf{x}_i, \bar{\mathbf{y}}_{N_i}^{(t)}; \Lambda^{(t-1)}) = \frac{P(\mathbf{x}_i, y_i|\bar{\mathbf{y}}_{N_i}^{(t)}; \Lambda^{(t-1)})}{\sum_{y_i=1}^{K} P(\mathbf{x}_i, y_i|\bar{\mathbf{y}}_{N_i}^{(t)}; \Lambda^{(t-1)})}, \tag{13}$$

$$P(\mathbf{x}_i, y_i|\bar{\mathbf{y}}_{N_i}^{(t)}; \Lambda^{(t-1)}) = \frac{\psi_u(\mathbf{x}_i, y_i; \Theta^{(t-1)})\psi_p(y_i, \bar{\mathbf{y}}_{N_i}^{(t)}; \beta^{(t-1)})}{Z_{\mathbf{x}_i}(\Lambda^{(t-1)})}, \tag{14}$$

where $\Theta = \{\mu_{y_i}, \Sigma_{y_i}\}$ denote the parameters of unary potential. $Z_{\mathbf{x}_i}(\Lambda^{(t-1)}) = \sum_{y_i} \psi_p(y_i, \bar{\mathbf{y}}_{N_i}^{(t)}|\beta^{(t-1)})$ is the local partition function. Since $Z_{\mathbf{x}_i}$ is independent with y_i, it is eliminated in Eq. (13). Such that this problem becomes tractable.

M step: learning of Λ, *with fixed* $P(\mathbf{y}|X, \bar{\mathbf{y}}; \Lambda^{(t-1)})$

Λ is learned by

$$\underset{\Lambda}{\operatorname{argmax}} \sum_{\mathbf{y}} P(\mathbf{y}|\bar{\mathbf{y}}^{(t)}, \mathbf{x}_i; \Lambda^{(t-1)}) \log P(X, \mathbf{y}|\bar{\mathbf{y}}^{(t)}; \Lambda). \tag{15}$$

Specifically, we have

$$\underset{\Theta}{\operatorname{argmax}} \sum_{i=1}^{n} \sum_{y_i} P(\mathbf{y}|X, \bar{\mathbf{y}}; \Theta^{(t-1)}) \log \psi_u(y_i, \mathbf{x}_i; \Theta), \tag{16}$$

$$\underset{\beta>0}{\operatorname{argmax}} \sum_{i=1}^{n} \sum_{y_i} P(\mathbf{y}|X, \bar{\mathbf{y}}; \beta^{(t-1)}) \log \frac{\psi_p(y_i, \bar{\mathbf{y}}_{N_i}^{(t)}; \beta)}{Z_{\mathbf{x}_i}(\beta)}. \tag{17}$$

The Gaussian assumption of $\psi_u(y_i, x_i; \Theta)$ leads to the closed-form solution to $\Theta^{(t)}$, by setting the derivative as zero, as follows:

$$\boldsymbol{\mu}_k^{(t)} = \frac{\sum_{i=1}^{n} P(y_i = k|\bar{\mathbf{y}}_{N_i}^{(t)}, \mathbf{x}_i; \Lambda^{(t-1)})\mathbf{x}_i}{\sum_{i=1}^{n} P(y_i = k|\bar{\mathbf{y}}_{N_i}^{(t)}, \mathbf{x}_i; \Lambda^{(t-1)})}, \tag{18}$$

$$\Sigma_k^{(t)} = \frac{\sum_{i=1}^{n} P(y_i = k|\bar{\mathbf{y}}_{N_i}^{(t)}, \mathbf{x}_i; \Lambda^{(t-1)})\epsilon_{ik}\epsilon_{ik}^T}{\sum_{i=1}^{n} P(y_i = k|\bar{\mathbf{y}}_{N_i}^{(t)}, \mathbf{x}_i; \Lambda^{(t-1)})}, \tag{19}$$

where ϵ_{ik} represents $(\mathbf{x}_i - \boldsymbol{\mu}_k^{(t)})$ for short. For β, we find a local optimal value in each iteration through the local search method [42].

4.1.2. An efficient framework for face clustering

Constrained clustering often has a higher computational cost than standard clustering, as it requires to satisfy cluster assumption and constraints simultaneously. The number of detected faces in videos is often up to several thousands or more, which means a very high clustering cost. To alleviate this limitation, we propose an efficient algorithm for face clustering in videos. It is observed that faces in adjacent frames of the same tracklet are very similar in appearance. There is a large amount of information redundancy within one tracklet. Hence it is reasonable to choose a small subset from each tracklet and firstly perform clustering on the subset, then the cluster labels of the whole data set are determined based on the labels of subset. On the other hand, we should also preserve the appearance variability to avoid information loss. Such that a moderate size of the subset should be determined, by taking account of the size of the whole data, the data quality, as well as the requirements on computational cost and clustering accuracy. Specifically, we uniformly sample a fixed number of faces from each tracklet. Moreover, after obtaining the clustering labels of the sampled faces, the clustering label of each tracklet can be determined by voting. For example, if the labels of the sampled 5 faces are predicted as (3,1,2,3,3), then the label of this tracklet is determined as 3.

Since this sampling trick is independent with the specific clustering model, it can be combined with any clustering models to reduce the computational cost. Take Algorithm 2 as an example, its main computational cost lies on two parts: the computation of the posterior probability $P(y_i|\mathbf{x}_i, \bar{\mathbf{y}}_{N_i}^{(t)})$ (Eq. (13)) takes $O(nKr)$, where $r \le n$ is the number of neighbors of each example (i.e., the number of non-zero entries in each row of W); the cost of learning Σ is $O(Knd^2)$. Considering the iteration number T, the computational complexity of Algorithm 2 is $O(TKn(r + d^2))$. When adopting Algorithm 2 in Algorithm 3, the complexity reduces to $O(TKn_s(r_s + d^2) + n)$ with the subscript s indicating the corresponding value of the subset. In our experiments $\frac{n_s}{n} \in (\frac{1}{23}, \frac{1}{10})$, leading to the significant cost reduction.

Algorithm 3. An efficient clustering framework for face clustering in videos.

Input: the whole data set X, neighborhood system W
Output: labels of the tracklets $\mathbf{y}$
1: Construct a subset X^s by downsampling, and determine the constraint matrix of the subset, i.e., W^s;
2: Adopt a clustering algorithm on X^s with W^s, and predict their labels $\mathbf{y}^s$;
3: Determine the labels of tracklets $\mathbf{y}$ based on $\mathbf{y}^s$.

4.2. Tracklet linking: optimizing O with fixed y

This step is achieved with the following optimization:

$$O^\star = \underset{O \in \{0,1\}^{m \times m}}{\operatorname{argmax}} \log P(M|O) + \sum_{\mathbf{y}} \log P(\mathbf{y}, O), \tag{20}$$

which is simplified by dropping constant terms to yield

$$O^\star = \underset{O \in \{0,1\}^{m \times m}}{\operatorname{argmax}} \sum_{i=1}^{m} \sum_{j=1}^{m} O_{ij}[\lambda_1 M_{ij} + \beta\lambda_2(\mathbb{I}(y_i = y_j) - 1)] + \beta \sum_{i=1}^{m} \left(\log\left[\mathbb{I}\left(\sum_{j=1}^{m} O_{ij} = 1 \right) \right] + \log\left[\mathbb{I}\left(\sum_{j \neq i}^{m} O_{ji} \le 1 \right) \right] \right). \tag{21}$$

Eq. (21) can be considered as a matching problem of a weighted bipartite graph, which can be solved by the Hungarian algorithm [31]. Specifically, $[\lambda_1 M_{ij} + \beta\lambda_2(\mathbb{I}(y_i = y_j) - 1)]$ denotes the edge weight, and if this edge is selected, then $O_{ij} = 1$. Obviously in the setting of bipartite matching, the constraints embedded in $\mathbb{I}(\sum_{j=1}^{m} O_{ij} = 1)$ and $\mathbb{I}(\sum_{j \neq i}^{m} O_{ji} \le 1)$ are satisfied automatically. Note that if we set $\lambda_2 = 0$, then (21) reduces to a basic linking problem, without the help of clustering results.

Tracklet similarity: As shown in problem (21), the tracklet similarity matrix M plays the key role in tracklet linking. Following some previous works [29,30], the similarity takes account of three aspects, including the temporal adjacency, appearance affinity and motion smoothness. The overall similarity measure is formulated as follows:

$$M_{ij} = \begin{cases} e^{-d_t(\mathbf{t}^{(i)}, \mathbf{t}^{(j)}) - \eta_1 d_a(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) - \eta_2 d_m(\mathbf{l}^{(i)}, \mathbf{l}^{(j)})} & i \neq j \\ \xi & i = j, \end{cases} \tag{22}$$

where constants η_1 and η_2 are two trade-off parameters.

Specifically, distance d_t enforces the temporal constraint: *if* $\mathbf{u}_j$ *occurs before* $\mathbf{u}_i$ *or they are overlapped in some frames, then* $O_{ij} = 0$. Here $\mathbf{t}^{(i)} = (t_1^{(i)}, \ldots, t_{n_i}^{(i)})'$ is a column vector containing the frame indices

of all faces in tracklet $\mathbf{u}_i$ (see Table 1). Similar to the work in [30], we define d_t as:

$$d_t(\mathbf{t}^{(i)}, \mathbf{t}^{(j)}) = \begin{cases} 0, & \text{if } (0 < \Delta t_{ij} < t_0), \\ \infty, & \text{otherwise,} \end{cases} \tag{23}$$

where $\Delta t_{ij} = t_1^{(j)} - t_{n_i}^{(i)}$ indicates the temporal difference between $\mathbf{u}_i$ and $\mathbf{u}_j$, and t_0 is a pre-defined threshold to avoid linking two tracklets with a large frame gap.

d_a measures the appearance distance. The appearance of each detected face is represented as the vector of concatenating the RGB values of each pixel (the dimension will be reduced by PCA). A tracklet is further represented by the average vector of the included faces. Then the Euclidean distance is computed as $d_a(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$.

Last, distance d_m reflects the motion smoothness. Specifically, denote $\mathbf{l}_j^{(i)} \in \mathrm{R}^{4\times 1}$ as the location and scale of the jth bounding box in tracklet $\mathbf{u}_i$, represented by the horizontal and vertical coordinate of the central pixel, and the width and height of the box. Treating each face $\mathbf{l}_j^{(i)}$ as a point, then one tracklet can be seen as a sequence of points in a 4-dimensional space. We fit this sequence through the polynomial curve fitting and the fitted curve $\mathbf{s}^{(i)}(\cdot)$ can be used to predict the bounding boxes of the other tracklet $\mathbf{u}_j$. The difference between the predicted bounding box and the true box is utilized to define d_m, as follows:

$$d_m(\mathbf{l}^{(i)}, \mathbf{l}^{(j)}) = \sum_{r\in\{1,2,3\}} \| \mathbf{s}^{(i)}(t_r^{(j)}) - \mathbf{l}_r^{(j)} \| + \sum_{r\in\{n_i-2,n_i-1,n_i\}} \| \mathbf{s}^{(j)}(t_r^{(i)}) - \mathbf{l}_r^{(i)} \|, \tag{24}$$

where $\mathbf{s}^{(i)}(t_r^{(j)})$ denotes the predicted bounding box at frame $t_r^{(j)}$ on the curve $\mathbf{s}^{(i)}$ (see Fig. 4).

The diagonal value $M_{ii} = \xi$ serves as a threshold to determine when to end a track. If ξ is large, then many short tracks will be obtained; otherwise fewer but longer tracks will be presented. In our experiments, ξ is determined as 3 times of the mode value among the finite off-diagonal values in M. The ratio $\frac{\beta\lambda_2}{\lambda_1}$ is adjusted to control the relative weight between constraints and tracklet similarities. We initially set $\frac{\beta\lambda_2}{\lambda_1} = 0.1\xi$. Since the clustering results are expected to become more accurate as the iteration proceeds, we gradually increase $\frac{\beta\lambda_2}{\lambda_1}$ during the optimization process.

For example, we can start from a small increasing rate, like $\frac{\beta\lambda_2}{\lambda_1} \leftarrow \frac{\beta\lambda_2}{\lambda_1} \times 1.2$ after each iteration. If the changes of the tracklet linking and clustering results between consecutive iterations are small, we can use a larger increasing rate, such as 1.5. If the changes are very sharp and unstable, we can use a smaller increasing rate.

We realize that in the field of tracklet linking, some sophisticated models have been specifically designed for some particular scenarios, where some specific prior knowledge can be utilized, such as social grouping and motion map. Compared with these models, the model of tracklet linking part used in our framework is relatively simple, as we does not assume some particular scenarios. However, our main contribution to the tracking problem is the first proposal of using the cluster label information to help tracklet linking. As we will show in the later experiments, due to the negative constraints from cluster labels, the ID switch error could be significantly reduced, leading to more pure long trajectories. This prior knowledge can be used in other tracklet linking models, while the techniques used existing tracklet linking models (like the metric learning for each tracklet in [36]) also can be adopted into our model to further improve the performance. Besides, although we assume the number of clusters K as a priori for our clustering part, it is not a restriction for the tracklet linking part, as only the pairwise constraints between two clusters are utilized, rather the number K.

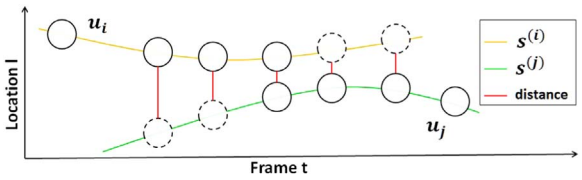

Fig. 4. Definition of d_m. The two central curves correspond to $\mathbf{s}^{(i)}$ and $\mathbf{s}^{(j)}$. The solid circles correspond to detected faces in one tracklet, while the dashed circles are those predicted by the fitted trajectory. We match detected faces that are highlighted in blue color. For simplicity, here we only show one dimension of $\mathbf{l}_j^{(i)}$, and the computations for the other three dimensions are similar. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

5. Neighborhood system

The neighborhood system of CHMRF captures all dependencies among the latent variables, as shown in Fig. 3-left and the pairwise potential functions defined in Section 3.3. Since there are two types of latent variables, i.e., $\mathbf{y}$ and O, the unified neighborhood system can be seen from two perspectives: one is the neighborhood system of $\mathbf{y}$, with fixed O, as shown in Fig. 3-middle; the other is the neighborhood system of O, with fixed $\mathbf{y}$, as shown in Fig. 3-right. As the second neighborhood system is fixed with given $\mathbf{y}$, hereafter we focus on how to generate the neighborhood system of clustering labels $\mathbf{y}$, i.e., the W matrix in Eq. (11).

In aforementioned definitions and formulations, each tracklet is treated as one sample of clustering. However, we find that there are unavoidable variations in appearance of the faces in the same tracklet, and in the lengths among different tracklets. As a result, it is difficult to describe one tracklet with respect to clustering. Actually, we treat each detected face as one sample of clustering, which can be naturally represented as a fixed length vector of the appearance of the bounding box. After clustering on each face, the cluster label of each tracklet can be easily determined (more details will be presented in Section 4.1.2). In the following we generate a neighborhood system among faces, represented as a $n \times n$ matrix W (see Eq. (11)) with $n = \sum_i^m n_i$ being the total number of detected faces. Note that in experiments we use the sampling trick (see Section 4.1.2) in the clustering part, thus n_i indicates the number of sampled faces from the tracklet i.

We present three approaches to obtain the neighborhood system: one is based on the data structure, and the obtained constraints are called as initial constraints, which are represented by the neighborhood system W_c; the second is based on two general assumptions, including *constraint consistency* and *constraint-level smoothness*, and the generated constraints are referred to as propagated constraints, which are embedded into W_{pc}; the last one is based on *example-level smoothness*, which leads to the neighborhood systems, including W_s and W_{com}.

5.1. Initial constraints

As the detected faces are organized as face tracklets, it is natural to explore pairwise constraints based on the relations among tracklets. In this work, we use constraints in the form of "must-link" and "cannot-link" [21,26] that are originated from the following two sources.

1. Spatiotemporal knowledge includes (a) based on temporal knowledge, faces in the same tracklet should belong to the same person, i.e., must-link, as shown in Fig. 5-left; (b) based on spatial knowledge, if two faces are detected in the same frame, then they should belong to different persons, i.e., cannot-link, as shown in Fig. 5-middle; (c) based on the transitivity of must-link, faces from two overlapped tracklets have "cannot-link" constraints, as shown in Fig. 5-right.
2. Tracklet linking results O. If two tracklets are linked, then the faces from this two tracklets are presumably of the same identity, which are then connected with the "must-link" constraint.

We save aforementioned constraints in a $n \times n$ matrix, as $W_c = V_0 + \lambda_2 \overline{O} \in \{-1, 0, \lambda_2, +1\}^{n\times n}$. Note that λ_2 is previously intro-

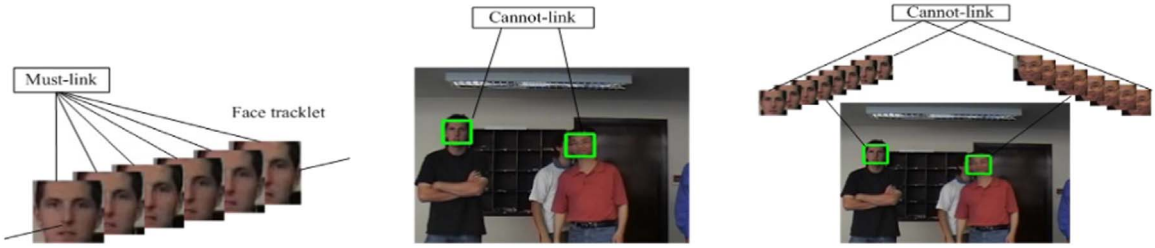

Fig. 5. Three types of spatiotemporal knowledge (faces are detected from Frontal [5]): (left) faces in the same tracklet are must-linked; (middle) two faces in the same frame are cannot-linked; (right) faces from two overlapped tracklets are cannot-linked.

duced in Eq. (7), and here it can be seen as the weight of the constraints from O. $V_0 \in \{-1, 0, +1\}^{n\times n}$ denote constraints from the spatiotemporal knowledge: given a must-link between the ith and jth face, then $V_0(i, j) = V_0(j, i) = +1$; for a cannot-link, $V_0(i, j) = V_0(j, i) = -1$; otherwise $V_0(i, j) = V_0(j, i) = 0$.$\overline{O} \in \{0, +1\}^{n\times n}$ represents the constraints from tracklet linking. Note that $O \in \{0, +1\}^{m\times m}$ denote the linking relations among tracklets, while $\overline{O}$ are pairwise constraints between cluster labels of faces. Given O, $\overline{O}$ can be uniquely determined. Specifically, if $O_{ij} = +1$, then $\overline{O}(r_i, r_j) = +1$ with r_i and r_j being the corresponding indexes of faces from the ith and jth tracklet respectively. Note that the non-zero entries in V_0 correspond to the faces from the same tracklet or two overlapped tracklets, while the ones in $\overline{O}$ correspond to the faces from two adjacent but non-overlapped tracklets. Such that the constraints from V_0 and $\overline{O}$ will not conflict with each other, and this is why $W_c(i, j) \in \{-1, 0, \lambda_2, +1\}$. Specifically, $W_c(i, j) > 0$ means a must-link between $\mathbf{x}_i$ and $\mathbf{x}_j$; $W_c(i, j) < 0$ represents a cannot-link; $W_c(i, j) = 0$ indicates no constraint. Note that if $|W_c(i, j)| = \infty$, then it is a hard constraint. It means if this constraint is violated, then the probability of such clustering results decreases to 0. However, we set finite values to the constraints, which means the constraints are softly embedded into our model. The constraint weights can be adjusted through β (see Eq. (4)), which can be learned automatically, as shown in Section 4.1.1.

5.2. Constraint propagation

The initial constraints W_c is often too sparse to achieve good clustering performance. We propose to augment the constraints through constraint propagation, based on two assumptions:

1. *constraint consistency* means that if $W_c(i, j) \neq 0$, then the propagated constraints $W_{pc}(i, j)$ should be close to $W_c(i, j)$. It ensures that the constraints be consistent before and after the propagation;
2. *constraint-level smoothness* encourages that given a must-link (cannot-link) between $\mathbf{x}_1$ and $\mathbf{x}_2$, if $\|\mathbf{x}_2 - \mathbf{x}_3\|_2^2$ is very small (i.e., they are close to each other), then it is assumed that there is also a must-link (cannot-link) between $\mathbf{x}_1$ and $\mathbf{x}_3$ [22].

The constraint propagation is formulated as follows:

$$\min_{W_{pc}} \frac{\gamma}{2}[\mathrm{tr}(W_{pc}^T L W_{pc}) + \mathrm{tr}(W_{pc} L W_{pc}^T)] - \mathrm{tr}(W_c^T W_{pc}), \quad \text{s. t.} \quad \mathrm{tr}(W_{pc}^T W_{pc}) = n^2, \tag{25}$$

where $-\mathrm{tr}(W_c^T W_{pc}) = -\sum_{i,j} W_c(i, j) W_{pc}(i, j)$ serves as the loss function to encourage the constraint consistency. The second term $\mathrm{tr}(W_{pc}^T L W_{pc} + W_{pc} L W_{pc}^T)$ can be expanded as follows [43]:

$$\mathrm{tr}(W_{pc}^T L W_{pc}) = \sum_{k=1}^{n} \sum_{i,j}^{n} A(i, j) \left(\frac{W_{pc}(i, k)}{\sqrt{d_i}} - \frac{W_{pc}(j, k)}{\sqrt{d_j}} \right)^2, \tag{26}$$

$$\mathrm{tr}(W_{pc} L W_{pc}^T) = \sum_{k=1}^{n} \sum_{i,j}^{n} A(i, j) \left(\frac{W_{pc}(k, i)}{\sqrt{d_i}} - \frac{W_{pc}(k, j)}{\sqrt{d_j}} \right)^2. \tag{27}$$

The normalized graph Laplacian matrix $L = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$, with the similarity matrix $A(i, j) = \exp(-\frac{dis(\mathbf{x}_i, \mathbf{x}_j)}{\sigma_i \sigma_j})$[4] and the diagonal degree matrix $D(i, i) = d_i = \sum_j A(i, j)$, as well as the identity matrix I. Minimizing this term means that if two points are close to each other in the feature space (i.e., $A(i, j)$ is large), then the corresponding two rows/columns of W_{pc} should be similar. It captures the constraint smoothness. The trade-off parameter γ controls the influence between constraint consistency and smoothness. The constraint term $\mathrm{tr}(W_{pc}^T W_{pc}) = n^2$ is introduced to avoid the blow-up of the entries in W_{pc}. Note that the above two constraint smoothness terms were firstly used in [43], where the constraint propagation was interpreted as a two-class semi-supervised learning problem, based on these two smoothness terms. Here we try to give a more clear interpretation. Treating each entry of W_{pc} as a node, and the value of $W_{pc}(i,j)$ is seen as the (soft) label of one node. Now we have a square graph with n^2 nodes, arranged in n rows and n columns. Then, the edges between nodes can be constructed from the instance similarity (i.e., the A matrix). Note that the edges only exist between the nodes in the same row and the nodes in the same column. Given this graph, the node label (i.e., the pairwise constraint) propagation can be understood that it consists of two parts: one is the propagation among the nodes in the same column (corresponding to Eq. (26); the other is the propagation among the nodes in the same row column (corresponding to Eq. (27)).

Problem (25) can be solved by *projected gradient descent* [45], which consists of two iterative steps, including *gradient descent* and *projection*, as follows.

Gradient descent : Denote the objective function without the constraint in Eq. (25) as $J(W_{pc})$, then its gradient w.r.t. W_{pc} is computed as follows:

$$\nabla W_{pc} = \gamma(W_{pc} L + L W_{pc}) - W_c. \tag{28}$$

Note that if we set $\nabla W_{pc} = 0$, then it leads to a Lyapunov equation [46]. It has been proved [47] that there is a unique solution of Lyapunov equation. However, for large matrices W_{pc} and L, directly solving the Lyapunov equation with matrix inversion is not very efficient. Hence, we resort to the efficient gradient descent method. Utilizing ∇W_{pc}, W_{pc} is updated in the $t+1$ iteration as follows:

$$W_{pc}^{t+1} = W_{pc}^t - \alpha^t \nabla W_{pc}^t, \tag{29}$$

where the step size α^t is determined by exact line search:

$$\alpha^t = \arg\min_{\alpha \in \mathbb{R}^+} J(W_{pc}^t - \alpha \nabla W_{pc}^t), \tag{30}$$

which is a simple convex optimization problem, leading to

$$\alpha^t = \frac{\gamma[\mathrm{tr}(W_{pc}^T L \nabla W_{pc}) + \mathrm{tr}(W_{pc} L \nabla W_{pc}^T)] - \mathrm{tr}(\nabla W_{pc}^T W_c)}{\gamma[\mathrm{tr}(\nabla W_{pc}^T L \nabla W_{pc}) + \mathrm{tr}(\nabla W_{pc} L \nabla W_{pc}^T)] \times (\eta * t)}, \tag{31}$$

where we ignore the iteration index t from W_{pc}^t and ∇W_{pc}^t for clarity. $\eta > 0$ is introduced to ensure the convergence. In experiments, we

[4] Following the suggestion in [44], the local kernel size σ_i is determined as the distance from $\mathbf{x}_i$ to 7-th nearest neighbor, i.e. $\sigma_i = dis(\mathbf{x}_i, \mathbf{x}_{i,7})$. Here the Euclidean distance is adopted as dis.

Table 2
Statistics of different real-world videos. *whole* denotes the set of all detected faces from every tracklet, while *subset* indicates the set of sampled faces from every tracklet.

Data	Time (s)	Frame	Person	Tracklet	Overlapped tracklet	Face	Dimen.	Original constraints		Constraints after linking	
								Must-link	Cannot-link	Must-link	Cannot-link
Frontal-whole [5]	51	1277	4	43	98	4267	10 800	400 370	1 785 225	430 428	1 785 225
Frontal-subset						215	5	430	2450	630	2450
Turning-whole [5]	40	1007	4	50	96	2799	10 800	137 738	545 947	851 657	545 947
Turning-subset						250	5	500	2400	6050	2400
BBT0101-whole [6]	1373	32 977	5	182	140	11 525	10 800	678 930	1 134 406	2 027 131	1 134 406
BBT0101-subset						546	10	546	1260	3021	1260
BBT0107-whole	1273	30 523	5	198	106	10 301	10 800	444 796	693 657	880 442	693 657
BBT0107-subset						594	10	594	954	1692	954
Notting-Hill-whole [49]	7442	178 439	7	277	86	19 278	10 800	1 103 449	758 868	1 659 416	758 868
Notting-Hill-subset						831	10	831	774	1578	774

gradually increase η in the set $\{0.5, 1, 2, 3, 5, 10\}$, until both $J(W_{pc})$ and α get convergence, in which case the obtained clustering results are always satisfied.

Projection: The updated W_{pc}^{t+1} is further projected into the constraint space:

$$W_{pc}^{t+1} \leftarrow W_{pc}^{t+1} / \sqrt{\mathrm{tr}((W_{pc}^{t+1})^T W_{pc}^{t+1})/n^2}. \tag{32}$$

Note that in our previous works [4,8], we adopted another constraint propagation method, which was firstly proposed in [43]. In the old method, the ℓ_2 loss $\|W_c - W_{pc}\|_F^2 = \mathrm{tr}(W_c^T W_c + W_{pc}^T W_{pc} - 2W_c^T W_{pc})$ encourages the small values of the entries in W_{pc} after propagation, which is not reasonable. Besides, its approximate solution involves with matrix inversion. In contrast, the loss in (25) only captures the constraint consistency. And our exact solution only involves with matrix multiplication, which is much more efficient than matrix inversion.

5.3. Example-level smoothness

Besides above constraints, another type of label correlation is derived from the example-level smoothness: if two observations $\mathbf{x}_i$ and $\mathbf{x}_j$ are similar, then their labels y_i and y_j should also be similar. This can be seen as soft must-link constraints. Specifically, we adopt the normalized affinity matrix to embed such smoothness, i.e., $W_s = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$. Furthermore, a natural choice is to combine W_s and the above constraints, i.e., $W_{com} = W_{pc} + \varepsilon W_s$ with a trade-off parameter $\varepsilon > 0$. Its tuning will be demonstrated in the following experimental part.

6. Experiments

In this section we evaluate the proposed algorithm on several real videos, and compare with state-of-the-art methods in face clustering and face tracklet linking.

6.1. Videos, face detection and tracklet generation

Three different sets of real videos are tested, representing different challenges for face clustering and tracklet linking. The first set contains two short videos, Frontal and Turning [5]. Their backgrounds are fixed, but frequent intersections and occlusions exist, as well as many pose changes in Turning. The second set includes two episodes BBT0101 and BBT0107 [6], from the TV series Big Bang Theory (Episode 01-01 and 01-07). These two videos contain mostly indoor scenes, but there are frequent changes of camera view, pose and illumination. Besides, many small faces exist due to the full scene shots. The third is extracted from the movie Notting-Hill [48]. This video is the most challenging one in our experiments, due to significant variations in scene, camera view, illumination, pose, resolution.

Given an input video, we generate reliable face tracklets by firstly applying the Viola–Jones face detector [49] in each frame. The detected faces in adjacent frames are then linked, based on similarities in their appearances, locations and scales of the bounding boxes. The small tracklets including less than υ faces are deleted in our experiments. Specifically, $\upsilon = 10, 20, 30$ in the aforementioned three sets of videos respectively. Moreover, as there are many characters in Notting-Hill, and most of them only appear in few scenes. It is difficult to clustering all these characters. Instead we focus on seven main characters, including 'Anna', 'William', 'Bella', 'Honey', 'Max', 'Bernie' and 'Spike'. Note that in our previous work [4], only 5 main characters and 76 tracklets are extracted from the video Notting-Hill. That is why the reported clustering results between this manuscript and [4] are different on the same video.

Each face is scaled to a 60×60 image, and represented by a 10 800-dimensional vector through concatenating the RGB values of all pixels. However, such a high feature space is the big challenge in computational cost and clustering accuracy for all methods. We adopt PCA to project the original space to a low dimensional space. The detailed statistics of all data sets are presented in Table 2.

6.2. Face clustering

6.2.1. Experimental settings

We adopt two commonly used metrics to evaluate clustering results on all faces, including clustering accuracy (ACC) and normalized mutual information (NMI).[5] The larger values of both metrics correspond to better performances. Each algorithm is conducted 10 times on each data, and the mean and the standard deviation are computed as the outputs. The parameter tuning involved in the clustering part of CHMRF should be firstly demonstrated. The trade-off parameter between constraint consistency and smoothness γ in Eq. (25) is tuned as follows: $\frac{\gamma}{1+\gamma}$ is chosen from the range $\{0.1, 0.2, \ldots, 0.9\}$, and the one leading to larger value of the objective function (15) is preferred in our experiments. The search range of β is set as $\{1: 0.1: 3\}$. The parameter ε (see Section 5.3) is chosen from the range $\{10^{-4}, \ldots, 10^{-1}, 0.5, 1\}$. Besides, several state-of-the-art clustering methods of different types are also compared. Note that all constraints, including constraints from spatiotemporal knowledge and tracklet linking results are also applicable to these constrained clustering methods (but no iteration between

[5] The code is downloaded from "http://www.mathworks.com/matlabcentral/fileexchange/29047-normalized-mutual-information".

Table 3
Clustering results evaluated by both Accuracy and NMI on different videos. (Case 1: Individual clustering; Case 2: Clustering with linking. The best results in each row are highlighted in bold. See Section 6.2.1 for details.).

Data	Different cases	Accuracy (mean (std) %)								NMI (mean (std) %)							
		EKM [50]	CCL [22]	HK [26]	ULDML [2]	CH-s	CH-c	CH-pc	CH-com	EKM [50]	CCL [22]	HK [26]	ULDML [2]	CH-s	CH-c	CH-pc	CH-com
Frontal	Case 1	85.17 (5.64)	54.61 (0.00)	68.85 (0.00)	79.03 (0.00)	86.29 (0.00)	**94.45 (0.00)**	**94.45 (0.00)**	**94.45 (0.00)**	72.80 (4.31)	32.61 (0.00)	44.04 (0.00)	64.51 (0.00)	75.66 (0.00)	**89.08 (0.00)**	**89.08 (0.00)**	**89.08 (0.00)**
	Case 2	85.17 (5.64)	54.61 (0.00)	56.88 (0.00)	82.24 (0.00)	86.29 (0.00)	94.45 (0.00)	**95.08 (0.00)**	**95.08 (0.00)**	72.80 (4.31)	32.61 (0.00)	25.33 (0.00)	72.96 (0.00)	75.66 (0.00)	89.08 (0.00)	**91.21 (0.00)**	**91.21 (0.00)**
Turning	Case 1	47.22 (0.26)	67.85 (0.00)	**91.57 (0.00)**	56.56 (0.00)	47.09 (0.00)	47.65 (0.51)	80.85 (0.00)	80.85 (0.00)	30.57 (0.71)	69.73 (0.00)	**79.48 (0.00)**	47.38 (0.00)	30.25 (0.08)	30.62 (0.29)	61.58 (0.00)	61.58 (0.00)
	Case 2	47.22 (0.26)	97.75 (0.00)	**98.64 (0.00)**	79.85 (0.00)	47.09 (0.00)	97.75 (0.00)	97.75 (0.00)	97.75 (0.00)	30.57 (0.71)	93.48 (0.00)	**96.13 (0.00)**	62.99 (0.00)	30.25 (0.08)	93.48 (0.00)	93.48 (0.00)	93.48 (0.00)
BBT0101	Case 1	55.78 (1.63)	60.46 (0.00)	46.28 (0.00)	58.26 (0.00)	55.73 (2.40)	61.21 (5.06)	**70.20 (2.95)**	69.87 (4.26)	33.70 (3.25)	37.69 (0.00)	27.61 (0.00)	47.43 (0.00)	33.80 (2.71)	36.47 (5.77)	**50.17 (4.32)**	48.99 (6.21)
	Case 2	55.78 (1.63)	41.88 (0.00)	69.77 (0.00)	62.42 (0.00)	55.73 (2.40)	64.52 (2.61)	**74.02 (0.16)**	73.82 (1.60)	33.70 (3.25)	20.25 (0.00)	40.19 (0.00)	42.99 (0.00)	33.80 (2.71)	47.01 (5.46)	50.69 (1.17)	**52.86 (1.58)**
BBT0107	Case 1	49.78 (5.97)	50.46 (0.00)	43.74 (0.00)	64.58 (0.00)	51.18 (2.52)	61.15 (2.26)	**67.08 (1.49)**	66.29 (4.03)	34.73 (6.60)	33.98 (0.00)	22.76 (0.00)	49.95 (0.00)	36.32 (5.60)	46.83 (3.61)	**54.49 (2.24)**	52.75 (5.24)
	Case 2	49.78 (5.97)	62.70 (0.00)	41.81 (0.00)	69.21 (0.00)	51.18 (2.52)	71.09 (1.69)	74.70 (2.20)	**76.02 (0.05)**	34.73 (3.25)	42.35 (0.00)	20.09 (0.00)	53.20 (0.00)	36.32 (5.60)	58.03 (1.52)	60.42 (2.46)	**61.98 (0.03)**
Notting-Hill	Case 1	34.10 (3.44)	30.94 (0.00)	28.84 (0.00)	40.77 (0.00)	35.56 (2.24)	35.98 (2.80)	**44.69 (3.38)**	44.64 (1.73)	23.51 (4.34)	22.11 (0.00)	9.27 (0.00)	19.34 (0.00)	25.60 (5.09)	25.57 (1.73)	**31.07 (2.49)**	29.42 (2.41)
	Case 2	34.10 (3.44)	32.54 (0.00)	31.59 (0.00)	43.82 (0.00)	35.56 (2.24)	37.31 (4.95)	**47.94 (4.46)**	45.41 (5.54)	23.51 (4.34)	16.06 (0.00)	13.55 (0.00)	19.31 (0.00)	25.60 (5.09)	26.76 (3.37)	32.21 (1.49)	**32.29 (2.46)**

Table 4
Experimental results of tracklet linking on three videos. PT: number of predicted tracks. MT: mostly tracked tracks (larger is better). Frag: number of fragments (smaller is better). IDS: number of ID switch (smaller is better).

	Frontal [5]				Turning [5]				BBT0101 [6]				BBT0107				Notting-Hill			
Method	PT	MT	Frag	IDS	PT	MT	Frag	IDS	PT	MT	Frag	IDS	PT	MT	Frag	IDS	PT	MT	Frag	IDS
Roth et al. [6]	11	4	24	13	5	**4**	**6**	2	72	68	81	10	140	79	143	3	208	**182**	**213**	5
Basic-Linking	13	5	**20**	9	5	3	**6**	**1**	77	65	84	9	124	**87**	**128**	4	214	176	219	5
Unified-Linking	20	**5**	22	**3**	5	3	**6**	**1**	75	**69**	**80**	**7**	128	**87**	129	**1**	212	179	215	**3**

clustering and linking). (1) *Traditional clustering*: elliptical K-means (EKM) [50] is used as the baseline to measure to what extent the pairwise constraints can help the clustering. (2) *Constrained clustering*: constrained complete-link (CCL) [22] and HMRF-KMeans (HK for short) [26] are compared. No parameters of CCL need to be tuned, and it gives the fixed result. HK is implemented through the built-in function in WekaUT, and we download it from the website "http://www.cs.utexas.edu/users/ml/risc/code/". They are both adopted in Algorithm 3. (3) *Specific algorithm for face clustering in videos*: ULDML [2] is implemented using Matlab, and a part of the code is provided by the author Ramazan Gokberk Cinbis. It treats each tracklet as one sample, so it can be directly used for the whole data.

We present two cases of clustering results: (a) the clustering without the help of linking, i.e., $\lambda_2 = 0$, referred to as *Case* 1; (b) the clustering results of CHMRF, i.e., $\lambda_2 > 0$, referred to as *Case* 2. As demonstrated in Section 5, CHMRF can adopt different neighborhood systems of $\mathbf{y}$. To distinguish them, we denote the corresponding clustering methods as: CH-s with W_s; CH-c with W_c; CH-pc with W_{pc}; CH-com with W_{com}.

Note that compared with the clustering results reported in our previous conference paper [8], there are some differences. The main reason is that in this manuscript we propose a new neighborhood system for the clustering part, as demonstrated in Section 5. The influence of the change of the neighborhood system in the face clustering can be revealed from the comparison between the results of CH-pc in Case 1 (without the help of linking results) in Table 3 and the results of HMRF-pc in [8], on the same video. The accuracies of CH-pc are 94.45%, 80.85%, 70.2% on Frontal, Turning and BBT0101 respectively, while the corresponding accuracies of HMRF-pc are 94.95%, 67.83% and 59.61%. Except for the slight inferiority on Frontal, CH-pc shows much better results than HMRF-pc on both Turning and BBT0101, where the face clustering is difficult. This demonstrates the advantage of the new neighborhood system. There are also differences between the results of CH-pc and HMRF-pc in Case 2 (with the help of linking results). Except for the change of the neighborhood system, the other reason is the slight change in the linking results, which provides additional constraints for clustering. It will be demonstrated in Section 6.3.

6.2.2. Clustering results

Clustering results on Frontal are shown in the second row of Table 3. Most detected faces in this video are frontal faces, so it is easy to do clustering. In Case 1, the accuracies of CH-c, CH-pc and CH-com are up to 94.95%, which is higher than other methods by about 9.8–40%. In Case 2, as frontal faces have provided enough information, constraints from linking results only provide a few additional information, and the accuracy is slightly improved for CH-pc and CH-com, to 95.08%. The accuracy of ULDML increases, while the accuracy of HK decreases, and all others keep the same with Case 1. The evaluations by NMI are basically consistent with the evaluations by accuracy.

Clustering results on Turning are summarized in the third row of Table 3. As many profile faces are detected from this video, it is difficult for face clustering based on pure appearance information. The poor performances of EKM and CH-s have proved this point. In Case 1, the constraints from spatiotemporal knowledge help to improve the clustering performance, compared with EKM and CH-s. The accuracies of CH-pc and CH-com are 80.85%, while the accuracy of CH-c is only 47.65%. This demonstrates that the constraint propagation significantly augment the useful information contained in the initial constraints. HK shows the best performance, and CCL and ULDML give the poor performance. Obviously, except for HK, the performances of other methods are not satisfied. We believe the reason is that the spatiotemporal constraints cannot handle the scenario of drastic pose changes between a pair of non-overlapped tracklets. However, constraints from tracklet linking that take account of motion consistency will provide useful information to handle the difficulty of pose changes. As shown in Case 2, the performances of all constrained clustering methods are significantly improved. The accuracies of CH-c, CH-pc and CH-com are up to 97.75%. The evaluations by NMI are basically consistent. This example fully demonstrates that tracklet linking can help to improve the clustering performance, especially when drastic pose changes exist.

Clustering results on BBT0101 are presented in the fourth row of Table 3. In Case 1, CH-pc gives the highest accuracy 70.20%, which is higher than other methods by about 9–24%. In Case 2, the accuracy of CH-pc is further improved to 74.02%. The performances of other constrained methods are also improved. An exception is CCL, of which the performance in Case 2 decreases. The possible reason is CCL embeds hard constraints in the hard manner. However, constraints from tracklet linking may include errors, which may significantly harm the performance. The evaluations by NMI are basically consistent.

Clustering results on BBT0107 are presented in the fifth row of Table 3. In case 1, CH-pc gives the highest accuracy 67.08%, which is higher than others by about 2.5–23%. In Case 2, its accuracy is further improved to 74.70%, and CH-com shows the best performance of 76.02%. The performances of other constrained methods, except for HK, are also improved. This demonstrates that HK is not very robust to the constraint noises. The evaluations by NMI are basically consistent with the evaluations by accuracy.

Clustering results on Notting-Hill are shown in the last row of Table 3. In Case 1, the accuracy of CH-pc is 44.69%, which is higher than other compared methods by 4–12%. CCL and HK perform very poor on this data set, while ULDML performs much better. In Case 2, the accuracy of CH-pc is further improved to 47.94%, and the performances of other constrained clustering methods are also improved. The evaluations by NMI are basically consistent with the evaluations by accuracy. Note that compared with the above four videos, the movie Notting-Hill contains more frequent camera motions, scene and illumination changes, etc. It is more challenging for face clustering and tracklet linking. Such that the numerical values on Notting-Hill are lower than the ones on the above data sets. But the comparisons still verify the efficacy of the proposed methods.

The above comparisons lead to the following conclusions. (1) The proposed methods CH-pc and CH-com perform much better than other methods in most cases. (2) The constraints from the linking results can provide useful information to improve the clustering performances of the most constrained clustering methods. (3) CH-pc and CH-com are more robust to the constraint noises from the tracklet linking, while other methods may suffer from such noises. Such robustness may be derived from the constraint propagation and the learning of constraint

weights, which can soften the influence of the constraint noises. (4) the comparisons among CH-s, CH-c, CH-pc and CH-com demonstrate that the initial constraints, the constraint propagation and the weight learning of constraints make the key contributions to the performance of CHMRF, while the example-level smoothness can be seen as a small compensate.

6.3. Face tracklet linking

To evaluate the linking results, we adopt the following metrics used in [35]: the number of predicted tracks (PT, i.e., the long tracks after linking), mostly tracked tracks (MT, larger is better), Fragments (Frag, smaller is better) and ID switch (IDS, smaller is better). The parameters in tracklet linking are tuned as follows. η_1 and η_2 (see Eq. (21)) control the proportions of appearance and motion information in the tracklet similarity M. If the appearance information is discriminative, then we set $\eta_1 > \eta_2$, such as in Frontal, $\eta_1 = 0.9$ and $\eta_2 = 0.1$. If the motion is smooth, then we set $\eta_2 > \eta_1$, such as in Turning, we set $\eta_1 = 0.1$ and $\eta_2 = 0.9$. However, in more complex videos, it is difficult to decide which information is better, such as in other three videos, we set $\eta_1 = \eta_2 = 0.5$.

Similar with clustering, we compare the linking results in two cases: (1) $\lambda_2 = 0$, i.e., tracklet linking without the help of clustering, referred to as *Basic-Linking*; (2) $\lambda_2 > 0$, i.e., the linking result of CHMRF, referred to as *Unified-Linking*. Besides, we also compare with the state-of-the-art method in the literature of face tracklet linking [6].[6]

The tracklet linking results are presented in Table 4. Note that the outputs are unique values, because given the similarity matrix M and the cluster labels $\mathbf{y}$ of tracklets (derived from the clustering results of CH-com with the neighborhood system W_{com}), the Hungarian algorithm gives the global optimal solution. However, the simulated field algorithm finds the local optimum, and may give different cluster labels. In each iteration of Algorithm 1, we conduct the clustering process many times, then adopt the clustering corresponding to the highest value of the objective function (15) as the input of tracklet linking. Such that the output of tracklet linking of each iteration is unique.

As the clustering results provide negative constraints for tracklet linking, some incorrect links may be avoided. So the IDS of our unified model can be easily reduced, but with the possible minor increase of the Frag, as shown the results on Frontal and BBT0107. On BBT0101 and Notting-Hill, our unified model gives better results than Basic-Linking on all three measures. On Turning, the linking is good enough due to the smooth motions. So the clustering results fail to further improve the linking performance. Roth et al. [6] provides the state-of-the-art results. Our unified model achieves the similar results with [6] on Notting-Hill, and better results on the other four data sets.

Note that there are differences in the linking results in Table 4 in this manuscript and the reported linking results in our previous conference paper [8]. In terms of Basic-Linking, it only depends on the similarity between tracklets, i.e., the M matrix in Eq. (22). We use different values of the trade-off parameters η_1 and η_2 in this work and [8], as described in the above section. In terms of Unified-Linking, except for the result change of Basic-Linking, the change of the clustering results of CH-pc will also lead to the change of linking results.

7. Conclusion and discussions

We describe a novel framework that simultaneously clusters and associates faces of distinct humans in long video sequences for identity maintenance. We develop a Coupled Hidden Markov Random Field (CHMRF) model, in which the joint dependencies between cluster labels and tracklet linkings, the correlations among cluster labels and among linkings are simultaneously captured by the neighborhood system. Different prior knowledge, including spatiotemporal knowledge and constraint consistency/smoothness, are also exploited to augment constraints for clustering. These constraints are then naturally embedded into the neighborhood system. Based on CHMRF, we formulate the simultaneous clustering and linking problem as a Bayesian inference problem. An effective coordinate descent solution is presented, consisting of two iterative parts, i.e., simulated field algorithm for constrained clustering and Hungarian algorithm for tracklet linking. We show significant improvements on the state-of-the-art results in face clustering and tracklet linking on several challenging video data sets.

There are a few future directions we would like to further explore. In experiments we find that when significant occlusion or intersection occurs, both clustering and linking will suffer. It tells that the performance of the proposed model can be further improved by using more robust features and more sophisticated linking method. Furthermore, we will also investigate more efficient optimization procedures of the constrained clustering and matching problems, and incorporating the simultaneous face clustering and linking into an overall system for video summarization.

[6] The original method of [6] used cues from face poses. However, such cues are derived from the MCT face detector that can output the face pose and eye location. In our experiments, the input tracklets, which only tell the bounding box of each face, are the same for every method. To be the fair comparison, our implementation of [6] (its code is not directly available to us) is simplified to use only appearance and motion cues.

Acknowledgments

The work was completed when the first author was a visiting student at Rensselaer Polytechnic Institute (RPI), supported by a scholarship from China Scholarship Council (CSC). We thank CSC and RPI for their supports. Qiang Ji is supported in part by a grant from the US National Science Foundation (NSF, No. 1145152). Bao-Gang Hu and Baoyuan Wu are supported in part by the National Natural Science Foundation of China (NSFC, Nos. 61273196 and 61573348). We greatly thank Professor Siwei Lyu for his constructive comments to this work.

References

[1] Ying Li, CC Jay Kuo, Video content analysis using multimodal information: for movie content extraction, indexing and representation, Springer, 2003.
[2] Ramazan Gokberk Cinbis, Jakob Verbeek, Cordelia Schmid, Unsupervised metric learning for face identification in tv video, in: ICCV, 2011, pp. 1559–1566.
[3] Nicholas Vretos, Vassilios Solachidis, Ioannis Pitas, A mutual information based face clustering algorithm for movie content analysis, Image Vis. Comput. 29 (10) (2011) 693–705.
[4] Bao-Yuan Wu, Yifan Zhang, Bao-Gang Hu, Qiang Ji, Constrained clustering and its application to face clustering in videos, in: CVPR, 2013.
[5] Emilio Maggio, Elisa Piccardo, Carlo Regazzoni, Andrea Cavallaro, Particle phd filtering for multi-target visual tracking, in: ICASSP, vol. 1, 2007, pp. I-1101.
[6] Markus Roth, Martin Bauml, Ram Nevatia, Rainer Stiefelhagen, Robust multi-pose face tracking by multi-stage tracklet association, in: ICPR, 2012, pp. 1012–1016.
[7] D. Koller, N. Friedman (Eds.), Probabilistic Graphical Models: Principles and Techniques, MIT Press, Cambridge, MA, 2009.
[8] Bao-Yuan Wu, Siwei Lyu, Bao-Gang Hu, Qiang Ji, Simultaneous clustering and tracklet linking for multi-face tracking in videos, in: ICCV, 2013.
[9] Andrew W. Fitzgibbon, Andrew Zisserman, On affine invariant clustering and automatic cast listing in movies, in: ECCV, Springer, 2002, pp. 304–320.
[10] Andrew Fitzgibbon, Andrew Zisserman, Joint manifold distance: a new approach to appearance based clustering, in: CVPR, 2003, pp. I-26.
[11] Ruiping Wang, Shiguang Shan, Xilin Chen, Wen Gao, Manifold-manifold distance with application to face recognition based on image set, in: CVPR, 2008, pp. 1–8.
[12] Yiqun Hu, Ajmal S Mian, Robyn Owens, Sparse approximated nearest points for image set classification, in: CVPR, 2011, pp. 121–128.
[13] O. Arandjelovic, R. Cipolla, Automatic cast listing in feature-length films with anisotropic manifold space, in: CVPR, 2006, pp. 1513–1520.
[14] S. Prince, J. Elder, Bayesian identity clustering, in: Canadian Conference on Computer and Robot Vision, 2010, pp. 32–39.
[15] Ming Du, Rama Chellappa, Face association across unconstrained video frames using conditional random fields, in: ECCV, Springer, 2012, pp. 167–180.
[16] Lior Wolf, Tal Hassner, Itay Maoz, Face recognition in unconstrained videos with matched background similarity, in: CVPR, 2011, pp. 529–534.

[17] T.L. Berg, A.C. Berg, J. Edwards, M. Maire, R. White, Y. Teh, E. Learned-Miller, D. A. Forsyth, Names and faces in the news, in: CVPR, 2006, pp. 848–854.
[18] Shijie Xiao, Mingkui Tan, Dong Xu, Weighted block-sparse low rank representation for face clustering in videos, in: ECCV, Springer, 2014, pp. 123–138.
[19] Xiaochun Cao, Changqing Zhang, Chengju Zhou, Huazhu Fu, Hassan Foroosh, Constrained multi-view video face clustering, IEEE Trans. Image Process. 24 (11) (2015) 4381–4393.
[20] Xiaochun Cao, Changqing Zhang, Huazhu Fu, Si Liu, Hua Zhang, Diversity-induced multi-view subspace clustering, in: CVPR, 2015, pp. 586–594.
[21] K. Wagstaff, C. Cardie, S. Rogers, S. Schrdl, Constrained k-means clustering with background knowledge, in: ICML, 2001, pp. 577–584.
[22] D. Klein, S.D. Kamvar, C.D. Manning, From instance-level constraints to space-level constraints: making the most of prior knowledge in data clustering, in: ICML, 2002, pp. 307–314.
[23] Eric P Xing, Michael I Jordan, Stuart Russell, Andrew Ng, Distance metric learning with application to clustering with side-information, in: NIPS, 2002, pp. 505–512.
[24] N. Shental, A. Bar-Hillel, T. Hertz, D. Weinshall, Computing Gaussian mixture models with em using equivalence constraints, in: NIPS, 2004.
[25] Z.D. Lu, K.L. Todd, Penalized probabilistic clustering, Neural Comput. 19 (6) (2007) 1528–1567.
[26] Sugato Basu, Mikhail Bilenko, Arindam Banerjee, Raymond J. Mooney, Probabilistic semi-supervised clustering with constraints, In: Semi-supervised Learning, 2006, pp. 71–98.
[27] Cheng-Hao Kuo, Chang Huang, Ramakant Nevatia, Multi-target tracking by on-line learned discriminative appearance models, in: CVPR, 2010, pp. 685–692.
[28] Peter Nillius, Josephine Sullivan, Stefan Carlsson, Multi-target tracking-linking identities using Bayesian network inference, in: CVPR, volume 2, 2006, pp. 2187–2194.
[29] Bo Yang, Ram Nevatia, Multi-target tracking by online learning of non-linear motion patterns and robust appearance models, in: CVPR, 2012, pp. 1918–1925.
[30] Zhen Qin, Christian R. Shelton, Improving multi-target tracking via social grouping, in: CVPR, 2012, pp. 1972–1978.
[31] Harold W. Kuhn, The Hungarian method for the assignment problem, Nav. Res. Logist. Q. 2 (1–2) (1955) 83–97.
[32] Chang Huang, Bo Wu, Ramakant Nevatia, Robust object tracking by hierarchical association of detection responses, in: ECCV, Springer, 2008, pp. 788–801.
[33] Vivek Kumar Singh, Bo Wu, Ramakant Nevatia, Pedestrian tracking by associating tracklets using detection residuals, in: IEEE Workshop on Motion and video Computing, 2008, pp. 1–8.
[34] Bi Song, Ting-Yueh Jeng, Elliot Staudt, Amit K Roy-Chowdhury, A stochastic graph evolution framework for robust multi-target tracking, in: ECCV, Springer, 2010, pp. 605–619.
[35] Yuan Li, Chang Huang, Ram Nevatia, Learning to associate: Hybridboosted multi-target tracker for crowded scene, in: CVPR, 2009, pp. 2953–2960.
[36] Bing Wang, Gang Wang, Kap Luk Chan, Li Wang, Tracklet association with online target-specific metric learning, in: CVPR, 2014, pp. 1234–1241.
[37] Shu Zhang, Yingying Zhu, Amit Roy-Chowdhury, Tracking multiple interacting targets in a camera network, Comput. Vis. Image Underst. 134 (C) (2015) 64–73.
[38] Seung-Hwan Bae, Kuk-Jin Yoon, Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning, in: CVPR, 2014, pp. 1218–1225.
[39] Xuan Song, Xiaowei Shao, Huijing Zhao, Jinshi Cui, Ryosuke Shibasaki, Hongbin Zha, An online approach: learning-semantic-scene-by-tracking and tracking-by-learning-semantic-scene, in: CVPR, 2010, pp. 739–746.
[40] G. Celeux, F. Forbes, N. Peyrard, Em procedures using mean field-like approximations for Markov model-based image segmentation, Pattern Recognit. 36 (1) (2003) 131–144.
[41] George Casella, Edward I. George, Explaining the Gibbs sampler, Am. Stat. 46 (3) (1992) 167–174.
[42] R. Battiti, M. Brunato, F. Mascia, Reactive Search and Intelligent Optimization, Operations research/Computer Science Interfaces, vol. 45, Springer Verlag, 2008.
[43] Z.W. Lu, H.H.S. Ip, Constrained spectral clustering via exhaustive and efficient constraint propagation, in: ECCV, 2010, pp. 1–14.
[44] Z.M. Lihi, P. Perona, Self-tuning spectral clustering, in: NIPS, 2004, pp. 1601–1608.
[45] R. Fletcher, Practical methods of optimization, 2nd ed., Wiley-Interscience, New York, NY, USA, 1987.
[46] Richard H. Bartels, G.W. Stewart, Solution of the matrix equation ax+ xb= c [f4], Commun. ACM 15 (9) (1972) 820–826.
[47] Peter Lancaster, Explicit solutions of linear matrix equations, SIAM Rev. 12 (4) (1970) 544–566.
[48] Y.F. Zhang, C.S. Xu, H.Q. Lu, Y.M. Huang, Character identification in feature-length films using global face-name matching, IEEE Trans. Multimed. 11 (7) (2009) 1276–1288.
[49] Paul Viola, Michael Jones, Rapid object detection using a boosted cascade of simple features, in: CVPR, 2001.
[50] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, New York, 2006.

Baoyuan Wu received his B.S. degree in Automation from University of Science and Technology Beijing, China in 2009, and his Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences in 2014. From September 2011 to September 2013, he was a visiting student in Rensselaer Polytechnic Institute, Troy, NY, USA. He is currently a Post-Doctoral Research Associate in King Abdullah University of Science and Technology. His main research interests include probabilistic graphical models, multi-label learning and discrete optimization.

Bao-Gang Hu received his M.Sc. degree from the University of Science and Technology Beijing, China in 1983, and his Ph.D. degree from McMaster University, Canada, in 1993, all in Mechanical Engineering. From 1994 to 1997, Dr. Hu was a Research Engineer and Senior Research Engineer at C-CORE, Memorial University of Newfoundland, Canada. Currently, he is a Professor with NLPR (National Laboratory of Pattern Recognition), Institute of Automation, Chinese Academy of Science, Beijing, China. From 2000 to 2005, he was the Chinese Director of LIAMA (the Chinese-French Joint Laboratory for Computer Science, Control and Applied Mathematics). His main research interests include intelligent systems, pattern recognition, plant growth modeling. He is a Senior Member of IEEE.

Qiang Ji received his Ph.D. degree in Electrical Engineering from the University of Washington. He is currently a Professor with the Department of Electrical, Computer, and Systems Engineering at Rensselaer PolytechnicInstitute (RPI). From 2009 to 2010, he served as a program director at the National Science Foundation (NSF), where he managed NSF's computer vision and machine learning programs. He also held teaching and research positions with the Beckman Institute at University of Illinois at Urbana-Champaign, the Robotics Institute at Carnegie Mellon University, the Department of Computer Science at University of Nevada at Reno, and the US Air Force Research Laboratory. Prof. Ji currently serves as the director of the Intelligent Systems Laboratory (ISL) at RPI. Prof. Ji's research interests are in computer vision, probabilistic graphical models, pattern recognition, and their applications in various fields. He has published over 200 papers in peerreviewed journals and conferences, and he has received multiple awards for his work. His research has been supported by major governmental agencies including NSF, NIH, DARPA, ONR, ARO, and AFOSR as well as by major companies. Prof. Ji is an editor on several related IEEE and international journals and he has served as a general chair, program chair, technical area chair, and program committee member in numerous international conferences/workshops. Prof. Ji is a fellow of the IEEE and the IAPR.