A multi-cue spatio-temporal framework for automatic frontal face clustering in video sequences

Siméon Schwab, Thierry Chateau, Christophe Blanc, Laurent Trassoudaine

EURASIP Journal on Image and Video Processing
a SpringerOpen Journal

RESEARCH Open Access

A multi-cue spatio-temporal framework for automatic frontal face clustering in video sequences

Simeon Schwab[1,2*], Thierry Chateau[1], Christophe Blanc[1,2] and Laurent Trassoudaine[1]

Abstract

Clustering of specific object detections is a challenging problem for video summarization. In this article, we present a method to form tracks by grouping face detections of a video sequence. Our clustering method is based on a probabilistic maximum *a posteriori* data association framework, and we apply it to face detection in a visual surveillance context. Optimal solution is found with a procedure using network-flow algorithms described in previous pedestrian tracking-by-detection works. To address difficult cases of small detections in scenes with multiple moving people, given that face detections are located in a video sequence, we use dissimilarities involving appearance and spatio-temporal information. The main contribution is the use of an optical flow or local front–back tracking to handle complex situations appearing in real sequences. The resulting algorithm is then able to deal with situations where people are crossing one another and face detections are scattered due to head rotation. The clustering step of our framework is compared to generic clustering methods (hierarchical clustering and affinity propagation) on several real challenging sequences, as evaluations indicate that this is more adapted to video-based detection clustering. We propose to use a new evaluation criteria, derived from purity and inverse purity of a clustering estimation, to assess performances of such methods. Results also show that optical flow and a skin color prior added to face detections improve the clustering quality.

Keywords: Clustering, Face detection, Multiple visual tracking, Optical flow, Maximum *a posteriori*

1 Introduction

Face detection on still images is becoming more and more common and efficient, yet use in real surveillance video sequences remains a big issue. Due to the large number of detections extracted from video, an automatic clustering of face detections is interesting for visual surveillance applications. For archive browsing or for face tagging on videos, it is easier to investigate with an album of faces than with a set of all the detected faces.

We propose a method to cluster-specific object detections of a video sequence, which we applied to face detections. Our efforts focus on real visual surveillance constraints: cluttered scenes, uncontrolled, and containing multiple small faces.

*Correspondence: simeon.schwab@univ-bpclermont.fr
[1]Institut Pascal, Université Blaise Pascal, 24, avenue des Landais, 63177 Aubière Cedex, France
[2]Vesalis, Parc Technologique de la Pardieu, 8, allée Evariste Galois, 63000 Clermont-Ferrand, France

In uncontrolled visual surveillance scenes, the use of a face recognition system remains complicated due to the poor quality of face images. It is for this reason that our method focuses on grouping face detections extracted from a video sequence and we do not address directly the face recognition problem. Our goal is to form tracks of face detections occurring in a whole video sequence.

The proposed method is based on three main stages (cf. Figure 1). First we use a face detector to localize faces in all frames of the video and extract various features of the detections. These features are then used to compute a dissimilarity matrix based on appearance and space–time. Finally, an optimization method involving a probabilistic model is employed to group all the detections according to the dissimilarity matrix.

2 Related work

Actual face tagging systems present interesting results on TV shows/series or news videos. The main works [1-9]

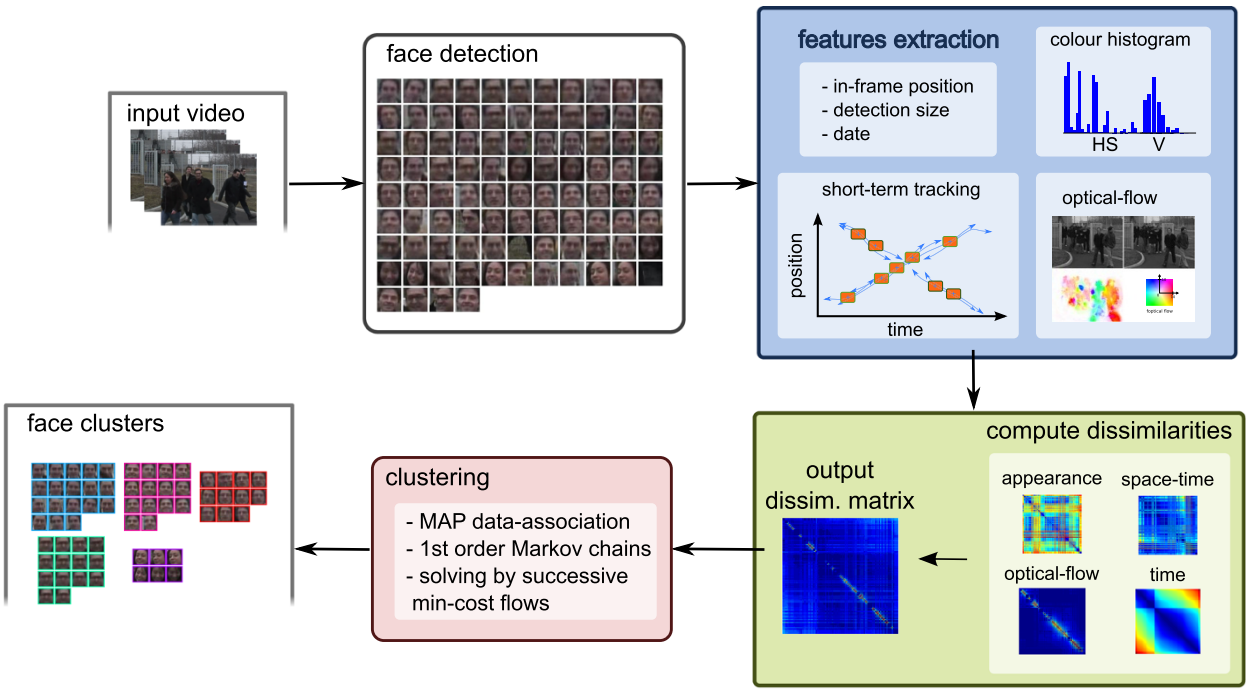

Figure 1 First, faces are extracted from each video frame by a standard face detector. Then appearance features and spatio-temporal features are built to compute different dissimilarity matrices. Finally, these dissimilarity matrices are combined to a single one and clustering is performed by solving a MAP estimation problem.

combine: a face detector in still image, face tracking techniques, and face recognition system with different probabilistic frameworks.

Some of them are semi-supervised: the study by Ramanan et al. [1] proposes a pre-clustering step to drastically reduce the hand-labeling time, Berg et al.'s study [2] automatically corrects inaccurately and ambiguously labeled face from news videos. In most of cases studied, a recognition system can be used, thanks to the detection quality and external information is sometimes added. Sivic et al. [4] work with near frontal faces and use other attributes such as hair or clothes to describe a person, and an extension of this method [3] uses subtitles to improve labeling and naming of persons.

All of these works use news, TV shows/series videos, as materials to face tagging in videos. However, in surveillance video scenes, people are not looking at the camera, close-up face views are rare, and a lot of mutual occlusions occurs. These are some of the reasons why face clustering remains extremely challenging in unconstrained visual surveillance situations.

Recent works on multi-object tracking can be seen as partial solutions to face clustering in video sequences. Such methods are generally divided into two main parts: (1) object detection and (2) data association. In these multi-object trackers, the data association problem is crucial having to handle complex situations such as partial or total occlusion. In fact, the problem of multi-object tracking and clustering of detected moving objects has many similarities to our problem. Many works in multi-object tracking have proposed some partial solution [10-13] using sequential or global strategies to estimate object trajectories and overcome identification errors.

To track different objects we often have to extrapolate the trajectories to predict the tracking during occlusions. To improve the quality of multi-object tracking one way is to add space–time information, and one currently used solution is short-term tracking. The studies [14-16] use a single object tracker with various probabilistic models and sampling.

For global tracking-by-detection, the main problem is to find a solution in a non-prohibitive computation time, because of the complexity of the possible detection combinations. To find a solution of a data association problem, as occurring in multi-target tracking, a lot of specific clustering algorithms are also employed: sampling with Monte Carlo methods [14,15,17], optimization methods like linear programming [18], Hungarian algorithm [19,20], or successive min-cost flow [21] on a graph. We chose to use this last method, and find a solution to the maximum *a posteriori* (MAP) by successive searches of min-cost flows on a graph [21]. Even though it restricts the problem model, this method is advantageous in that it finds an optimal solution in a reasonable computation time.

In this article, we employ the problem definition and solving of [21]. To add space–time cue, we extended face detections with forward and backward tracking. However, most of the time, short-term object trackers cannot actually handle occlusions as they are just used to estimate the motion of the object to fill time gaps. That is why, our idea is to use optical flow to add a local space–time information. We also tested the use of a generic clustering method for the third stage of our method. As the dissimilarities employed in visual detection clustering are not necessarily metrics, we focus on non-metric clustering methods. We selected two clustering methods from all the algorithms available. The first is a classical and famous method: hierarchical ascendant clustering. This method is an agglomerative clustering building a hierarchy according to dissimilarities between objects, this hierarchy could then be thresholded to define clusters. The other selected algorithm is the affinity propagation [22-24]. This algorithm performs unsupervised classification by identifying a subset of representative exemplars, and is also employed in a visual understanding context.

3 Probabilistic data-association model

This section presents the background we employed to define the probabilistic model used to cluster-detected faces. It is inspired by the model proposed by Nevatia and colleagues [21], we modified its formulation to clarify the likelihood and *a priori* probability terms and adapted it to the video-based face clustering case. To obtain an optimal clustering according to our probabilistic MAP framework, we used their network-flow-based algorithm.

3.1 MAP data-association model

The model is based on a probabilistic framework where a state (set of associations) has to be estimated from observations given by the set of faces extracted from a video sequence. State and observations are random variables, and the objective is to obtain MAP estimate of face associations.

Each observation (face detection) contains position on the frame, time in the sequence (frame index), appearance of the detection area, and motion information (detailed in the next section). Let $\mathcal{Z} = \{\boldsymbol{z}_i\}$ be the set of all the detections, where $\boldsymbol{z}_i = (\boldsymbol{x}_i, s_i, \boldsymbol{a}_i, t_i)$ is a detection; $\boldsymbol{x}_i$ represents the xy-position in pixels, s_i its width in pixels, $\boldsymbol{a}_i$ the appearance descriptor, and t_i the time (frame index) in the video. We denote D the number of detection in $\mathcal{Z}$.

The state to be estimated is a set of trajectories $T = \{T_1, \ldots, T_K, T_{\text{FP}}\}$, where a trajectory is a set of detections: $T_k = \{\boldsymbol{z}_{k_1}, \ldots, \boldsymbol{z}_{k_{n_k}}\}$ ($k_i = j$ means that the ith element of the k trajectory is the detection $\boldsymbol{z}_j$). The cluster denoted by T_{FP} represents the set of detections considered as false positives. As one detection can only belong to one trajectory and each detection is assigned to a trajectory, T is in fact a clustering of the D detections. $\mathcal{T}$ denotes the set of all the possible clustering of D detections.

The MAP estimation problem is then described as

$$\hat{T} = \arg\max_{T \in \mathcal{T}} P(T|Z) = \arg\max_{T \in \mathcal{T}} P(T)P(Z|T) \tag{1}$$

where $\mathcal{T}$ is the set of all the possible clustering of D detections. With independence hypothesis

$$\hat{T} = \arg\max_{T \in \mathcal{T}} P(T) \prod_i P(\boldsymbol{z}_i|T) \prod_k P(Z|T_k) \tag{2}$$

In [21], the detection likelihood ($P(\boldsymbol{z}_i|T)$) is represented with a Bernoulli distribution with a constant parameter. This parameter corresponds to the false positive rate of the face detector. We chose to delegate this term to the prior because it does not involve any observation, and we introduced a detection likelihood based on observations. This likelihood is described by the probability that an observation $\boldsymbol{z}_i$ is a real face and not a false positive of the detector. A more detailed description of this probability is given in Section 5.

As we delegate the false positive prior to the *a priori* probability, our *a priori* probability does not only involve the number of clusters as in [21], but also the number of detections considered as false positives. Assuming independence between the number of clusters and the number of false positive detections, the prior is defined as follows

$$\begin{aligned} P(T) &= P_{\text{start}}(K) P_{FP}(|T_{FP}|) \\ &= P_e^{2(K-1)} \beta^{|T_{FP}|} (1-\beta)^{D-|T_{FP}|} \end{aligned} \tag{3}$$

where P_e represents the probability of starting a trajectory with a given detection (estimated as the number of people over the number of detections), K is the number of trajectories in T and $|T_{FP}|$ the number of detections considered as false positives. The β parameter is the false positive rate of the detector, and D denotes the total detection count.

The trajectory likelihood $P(Z|T_k)$ represents the appearance and space–time consistency of a trajectory T_k. It is expressed by a first-order Markov chain, where states are the detections of T_k:

$$\begin{aligned} P(Z|T_k) &= P_{\text{link}}(\boldsymbol{z}_{k_1}|\boldsymbol{z}_{k_0}) P_{\text{link}}(\boldsymbol{z}_{k_2}|\boldsymbol{z}_{k_1}) \ldots \\ & \quad P_{\text{link}}(\boldsymbol{z}_{k_{l_k}}|\boldsymbol{z}_{k_{l_k-1}}) \end{aligned} \tag{4}$$

Section 6 describes how we used dissimilarities (involving time, appearance, and movement) to define the P_{link} probabilities.

In our case, we just used the following two parameters to describe a situation: the rate of false positive β and the

probability of the detector P_e to start a trajectory. These parameters could be statistically estimated with

$$\beta = \frac{\text{False positives}}{\text{Number of detection}}$$

$$P_e = \frac{\text{Number of people}}{\text{Number of detection}}$$

As the face detector fails when the camera is not in front of the face, missed detections are not necessarily attributed to occlusion events and an explicit occlusion model (as in [21]) is not very suitable.

It is therefore complicated to set up the P_e parameter, however, experiments at Section 7.3 show that, in practice, results are quite stable with the P_e variation, then we empirically set it to 1%.

3.2 Resolving the MAP

Computing all the solutions of the MAP model is in general a challenging task, mainly because of the computational complexity.

By using the first-order Markov chain hypothesis for the trajectory likelihood, the solution is tractable and an optimal solution to the MAP can be computed by successively solving min-cost flow problems on a specific graph [21]. Nodes of this graph represent detections and a path on the oriented graph represents a cluster. The cost of an arc between two nodes is assigned to computed transition dissimilarity (opposite of log-likelihood) between the corresponding two detections. So, with a given value, the min-cost flow determines the associations to be used for clustering. An optimal MAP solution is found by varying the flow value and iteratively solving min-cost flow problems.

4 Detection extension with movement information

In this section, we present the manner in which we introduce space–time information to face detections. Two ways of representing space–time cues are proposed: (1) a basic short-term tracker and (2) an optical flow estimation. The two approaches are compared in Section 7.3.2.

4.1 Short-term tracking

To overcome hard situations due to long periods of undetected faces, we first propose to extend detections by using short-term backward and forward tracking. This algorithm provides additional space–time information to help in situations involving two detections distant in time. The tracking system is based on the estimation of the optimal position and size of the reference face (which is the patch of the detection in the present frame) at a further frame. The optimization procedure uses a cost function based on the appearance dissimilarity with the reference patch. Search domain is bounded by priors: a maximum velocity and scale factor. The optimization core is achieved by a Nelder–Mead method based on simplexes with the previous estimation as starting point. Tracking is achieved for each detection, and pursuing in the past and future according to video time.

Introducing new space–time information with short-term tracking implies extending the observation z_i of a detection with the new positions given by the tracker. We denote this new observation (called *tracklet*) as

$$\tilde{\boldsymbol{z}}_i = \{\boldsymbol{z}_i^1, \boldsymbol{z}_i^2, \ldots, \boldsymbol{z}_i^{N_i}\} \tag{5}$$

where $\boldsymbol{z}_i^k = (\boldsymbol{x}_i^k, s_i^k, t_i^k)$ are positions and sizes estimated with the short-term tracking and where N_i is the number of elements in the tracklet of the detection i. The $\boldsymbol{z}_i^k$ are sorted by frame time and one of these is the previously defined $\boldsymbol{z}_i$.

4.2 Optical flow

Another way to take into account space–time information is to use optical flow. There are many methods to compute an optical flow between two frames, one of the main issues is to represent a large scale of displacements [25-27]. In our case, we used a pyramidal version of the Lucas–Kanade algorithm to represent both small and large displacements as explained in [26].

In each frame having a detection, we compute the optical flow from the previous to the current frame and from the current to the next frame, then these two optical flows are averaged. The resulting speed vector of a detection is obtained by taking the most representative flow vector in the detection area. This vector is added to the observation $\boldsymbol{z}_i$.

5 Detection likelihood

In our MAP framework, the detection likelihood explains the fact that a detection is a true or false positive of the detector. In the case of a color video, we propose adding this information with the skin color proportion of a detection. We define the likelihood as follows

$$P(\boldsymbol{z}_i|T) = \begin{cases} 1 - P_f(a_i) & \text{if } \boldsymbol{z}_i \in T_{\text{FP}} \\ P_f(\boldsymbol{a}_i) & \text{else} \end{cases} \tag{6}$$

where the probability to be a face (P_f) with an appearance $\boldsymbol{a}_i$ is estimated by the proportion of skin color pixels over the detected face patch. The skin color segmentation is simply done by fixed colorimetric boundaries [28]. Due to ethnic skin differences and colorimetric noise, skin color detection is far from being the best representation of skin proportion, but it still adds information in the main cases. To limit the exclusion of true detection without skin color pixels, we threshold $P_f(\boldsymbol{a}_i)$ to be 0.01 at minimum instead of 0.

Another way to improve the detector is to add the prior that a detection is on the foreground by using background extraction techniques. The drawback of this method is that it cannot handle moving objects such as pedestrian clothes or vehicles, and background extraction is complicated with non-fixed cameras.

6 Detection dissimilarities

This section describes the probability P_{link} to transit from one detection to another, this probability is based on dissimilarities between detections. This transition probability combines four affinities: appearance (A_{a}), motion(A_{m}), speed (A_{s}), and time (A_{t}):

$$P_{\text{link}}(\boldsymbol{z}_j|\boldsymbol{z}_i) = A_{\text{a}}(\boldsymbol{z}_i,\boldsymbol{z}_j)A_{\text{m}}(\boldsymbol{z}_i,\boldsymbol{z}_j)A_{\text{s}}(\boldsymbol{z}_i,\boldsymbol{z}_j)A_{\text{t}}(\boldsymbol{z}_i,\boldsymbol{z}_j) \quad (7)$$

Dissimilarities are integrated into the MAP framework as the opposite of the log-likelihood of a transition (cf. Section 3.1). The dissimilarity expression is then defined as follows

$$\begin{aligned} d(\boldsymbol{z}_i,\vec{z}_j) &= -\log(P_{\text{link}}(\boldsymbol{z}_j|\boldsymbol{z}_i)) \\ &= -\log(A_{\text{a}}(\boldsymbol{z}_i,\boldsymbol{z}_j)) - \log(A_{\text{m}}(\boldsymbol{z}_i,\boldsymbol{z}_j)) \\ &\quad -\log(A_{\text{s}}(\boldsymbol{z}_i,\boldsymbol{z}_j)) - \log(A_{\text{t}}(\boldsymbol{z}_i,\boldsymbol{z}_j)) \\ &= \tilde{d}_{\text{a}}(\boldsymbol{z}_i,\boldsymbol{z}_j)^2 + \tilde{d_{\text{m}}}(\boldsymbol{z}_i,\boldsymbol{z}_j)^2 \\ &\quad +\tilde{d_{\text{s}}}(\boldsymbol{z}_i,\boldsymbol{z}_j)^2 + \tilde{d}_{\text{t}}(\boldsymbol{z}_i,\boldsymbol{z}_j)^2 \end{aligned} \quad (8)$$

with $\tilde{d}_{\text{a}}$ the appearance dissimilarity, $\tilde{d_{\text{m}}}$ the motion dissimilarity, and $\tilde{d}_{\text{t}}$ the temporal dissimilarity.

All of these dissimilarities are reduced to be combined in a normalized way

$$\tilde{d}_x(\boldsymbol{z}_i,\boldsymbol{z}_j) = \frac{d_x(\boldsymbol{z}_i,\boldsymbol{z}_j)}{\sigma_x} \quad (9)$$

where x is a, m, s, or t and σ_x is the standard deviation estimated with all the $d_x(f,g)$ with $(f,g) \in \mathcal{Z} \times \mathcal{Z}$. The next sections describe the different d_x dissimilarities.

6.1 Appearance dissimilarity

Detection appearance is represented by an HS-V histogram [29]. This histogram is the concatenation of a 2D HS histogram and a 1D V histogram of image pixels (where H, S, and V represent hue, saturation, and value of a color, respectively). If the S and V values are large enough for a pixel, the pixel is counted in the HS histogram, or else it is counted in the V histogram. To measure the dissimilarity between two HS-V histograms, we used the Bhattacharyya coefficient.

By considering only the face detection area, color information is not sufficient to distinguish two different faces. We therefore extended the face detection to an area under the head, in order to retrieve color information from the pedestrian clothes, this is done by doubling down the detection area.

6.2 Space–time dissimilarities

For dissimilarities involving position in frame and frame time, we define four dissimilarities: two for the motion (based on tracklets or optical-flows), one for the speed (in pixel per frame time), and one for the time. If detections are extended with tracklets ($\tilde{\boldsymbol{z}}_i$ instead of $\boldsymbol{z}_i$) the *tracklet dissimilarity* is employed, if not, the optical flow is estimated and the *optical flow dissimilarity* is used.

6.2.1 Tracklet dissimilarity

To quantify the space–time continuity between two tracklets, the end of the first tracklet is interpolated to the beginning time of the other tracklet, and vice versa. An Euclidean distance is then measured between the final extrapolated position of the first tracklet and the beginning of the second, and another distance is computed between the extrapolated position of the second tracklet and the last position of the first tracklet (cf. Figure 2). Extrapolation is done using a constant velocity model estimated with an average of finite differences of the last tracklet positions. To use this measurement with two single detections, we just set their speed to zero.

The motion dissimilarity between two trajectories is obtained by averaging the two acquired distances, as shown in Figure 2. This average is weighted by the number of positions used to estimate the speed. In practice, in order to account possible high accelerations, the number of finite differences used to estimate the speed is limited.

If the two tracklets overlap (i.e., at least one date in common), the motion dissimilarity is computed from the spatial average of position distances on common frames.

Given $T^{ij}_{\text{inter}} = \{t_i^1,\dots,t_i^{N_i}\} \cap \{t_j^1,\dots,t_j^{N_j}\}$ the frame intersection and assuming that $\boldsymbol{z}_i^1$ is before $\boldsymbol{z}_j^1$, the movement dissimilarity is defined by

- if no overlapping (i.e., $T^{ij}_{\text{inter}} = \emptyset$):

$$d_m(\tilde{\boldsymbol{z}}_i,\tilde{\boldsymbol{z}}_j) = \frac{K_i d_{\text{pos}}(\hat{\boldsymbol{z}}_i,\boldsymbol{z}_j^1) + K_j d_{\text{pos}}(\hat{\boldsymbol{z}}_j,\boldsymbol{z}_i^{N_i})}{K_i + K_j} \quad (10)$$

  where $\hat{\boldsymbol{z}}_i$ is the forward extrapolation of $\tilde{\boldsymbol{z}}_i$, $\hat{\boldsymbol{z}}_j$ the backward extrapolation of $\tilde{\boldsymbol{z}}_i$, K_i (resp. K_j) is the number of elements used to estimate speed from $\tilde{\boldsymbol{z}}_i$ (resp. $\tilde{\boldsymbol{z}}_j$). In practice, we take $K_i = \min(10, N_i)$ for our experiments.
- if overlapping:

$$d_m(\tilde{\boldsymbol{z}}_i,\tilde{\boldsymbol{z}}_j) = \frac{\sum_{t\in T^{ij}_{\text{inter}}} d_{\text{pos}}(\boldsymbol{z}_i^{k^i(t)},\boldsymbol{z}_j^{k^j(t)})}{|T^{ij}_{\text{inter}}|} \quad (11)$$

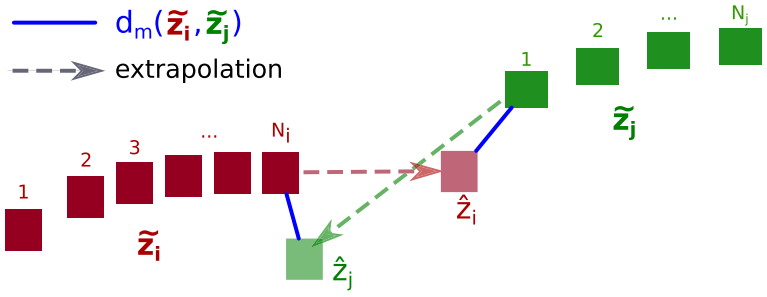

Figure 2 Movement dissimilarity between two non-overlapping tracks.

where $k^i(t_i^n) = n$ and d_{pos} are the position distance between two observations. This distance is an Euclidean distance divided by the mean of the two detection widths:

$$d_{\text{pos}}(z_i, z_j) = 2\frac{||\vec{x}_i - \vec{x}_j||}{s_i + s_j} \tag{12}$$

this normalization is done to be closer to a spatial overlap measure than the simple Euclidean distance is.

6.2.2 Optical flow dissimilarity

Instead of using tracklets, motion can also be estimated using an optical flow. The optical flow is computed by a pyramidal version of the Lucas–Kanade [26] algorithm, we use five levels for the image pyramids. Figure 3 shows an example of the estimated optical flow for a video of our dataset. Two optical flows are computed at each frame: one with the previous frame and the other with the next frame. The two obtained optical flows are then averaged to reduce the noise. The resulting velocity of a detection is estimated by the flow vector the most represented on the detection area.

The motion dissimilarity from optical flow is defined as follows

$$\begin{aligned} d_m(z_i, z_j) = \frac{1}{2}(&||\vec{x}_i + \vec{v}_i(t_j - t_i) - \vec{x}_j|| \\ &+||\vec{x}_j + \vec{v}_j(t_i - t_j) - \vec{x}_i||) \end{aligned} \tag{13}$$

where $\vec{v}_i$ is the estimated optical flow vector of the detection i.

6.2.3 Speed dissimilarity

Some associations are physically impossible due to the maximum speed that a person can achieve. To limit these associations, we use, in addition to the previous movement dissimilarity, a speed-based dissimilarity:

$$d_s(z_i, z_j) = \frac{||\vec{x}_i - \vec{x}_j||}{|t_i - t_j| + 1} \tag{14}$$

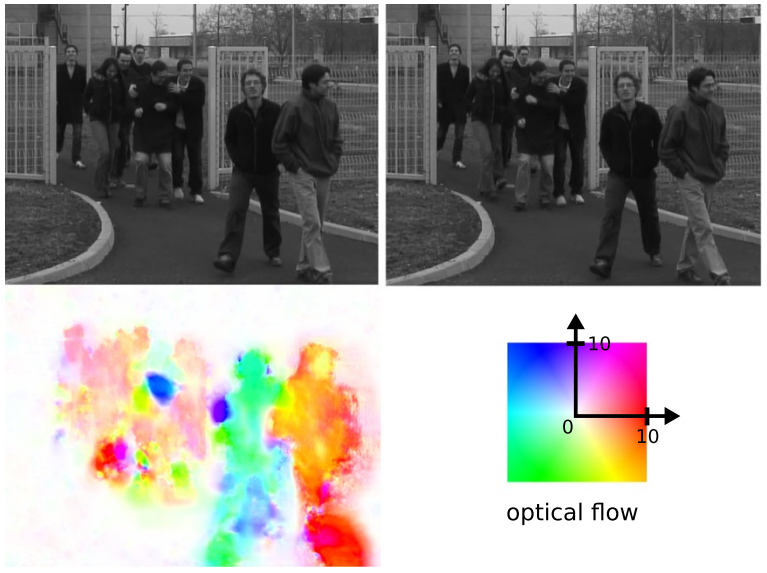

Figure 3 Optical flow computed with pyramidal Lucas–Kanade.

6.2.4 Time dissimilarity

The dissimilarity based on time is defined as the difference of times:

$$d_t(z_i, z_j) = \begin{cases} t_j - t_i & \text{if } \Delta t \geq t_j - t_i > 0 \\ \infty & \text{else} \end{cases} \quad (15)$$

where Δt is empirically set (50 frames for our experiments).

If tracklets are used, the dissimilarity is simply defined as the smallest gap between starts and ends of the two compared tracklets:

$$d_t(\tilde{z}_i, \tilde{z}_j) = \min(d_t(z_i^1, z_j^1), d_t(z_i^{N_i}, z_j^{N_j}), d_t(z_i^1, z_j^{N_j}), d_t(z_i^{N_i}, z_j^1)) \quad (16)$$

7 Evaluation

This section presents the evaluation of the proposed method on several challenging videos.

7.1 Evaluation criteria

There are several ways to measure clustering quality: intrinsic methods (by measuring the proximity of elements inside a cluster and the proximity between clusters) and extrinsic methods that use manual ground truth classification.

As shown by Amigó et al. [30], there are many ways to extrinsically evaluate clustering involving different quality measures, such as good and bad pair counting, purity, entropy measures, etc.

We propose an evaluation measurement based on purity and inverse-purity. We define *estimated clustering* as clustering obtained by a clustering algorithm, as opposed to *ground-truth clustering* manually achieved by a human expert. Moreover, the purity is called *estimation purity* (EP) and the inverse-purity *ground-truth purity* (GTP).

Assuming estimated clustering of all the detections, EP and GTP are defined as follows (cf. Figure 4):

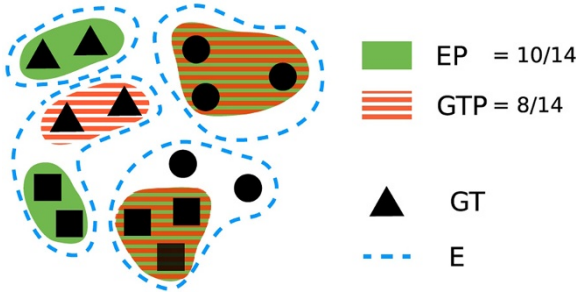

Figure 4 Illustration of EP and GTP.

- **EP**:

$$\text{EP} = \frac{1}{D} \sum_k \max_j |E_k \cap \text{GT}_j| \quad (17)$$

where D is the number of detections, $\text{GT} = \{\text{GT}_j\}$ the ground-truth clustering, and $E = \{E_k\}$ the estimated clustering. The higher the EP, the less there are covering errors. We refer to a covering error when a cluster includes the faces of different people.

- **GTP**:

$$\text{GTP} = \frac{1}{D} \sum_k \max_j |\text{GT}_k \cap E_j| \quad (18)$$

it shows the proportion of well-represented detections (i.e., it measures the fact that there are few people represented by multiple estimated clusters).

Considering the case of a single cluster gathering all the detections:

$$\text{EP} = \frac{\max_j |\text{GT}_j|}{N} (\text{noted } \text{EP}_{\min}) \text{ and } \text{GTP} = 1$$

and the case where each detection is represented by each cluster:

$$\text{EP} = 1 \text{ and } \text{GTP} = \frac{|\text{GT}|}{N} \text{ (noted } \text{GTP}_{\min})$$

we see that the minimum purities are not zero, but constants depending only on the number of detections and the largest cluster in the ground truth. To compare the various clustering algorithms, we made some changes to have the minimum of EP and GTP at 0.

$$\text{EP}_n = \frac{\text{EP} - \text{EP}_{\min}}{1 - \text{EP}_{\min}}$$

$$\text{GTP}_n = \frac{\text{GTP} - \text{GTP}_{\min}}{1 - \text{GTP}_{\min}}$$

The closer both EP and GTP are to 1, the better the clustering. So, we use an F-measure of EP_n and GTP_n to represent the quality by a single number between 0 and 1:

$$F = 2 \frac{\text{EP}_n \times \text{GTP}_n}{\text{EP}_n + \text{GTP}_n}$$

This measure is used in our experiments to compare the estimated clustering with the ground-truth clustering.

7.2 Experiments

7.2.1 Dataset

The experimentation dataset is based on Additional files 1, 2, 3, 4, 5 and 6 . Some figures are given in Table 1.

Videos (Additional file 1: Video 1, Additional file 2: Video 2, Additional file 3: Video 3, Additional file 4: Video 4, and Additional file 5: Video 5) constitute a challenging

Table 1 Dataset for experiments

Video	Passages	Frames	nb detect.	FP (%)	Size
1	24	1934	1725	2.78	35
2	6	307	200	11.5	35
3	7	384	920	2.61	36
4	6	485	463	3.46	58
5	29	1966	1794	1.56	63
6	14	5042	1299	22.17	32

Passages, number of people crossing the viewed area with a detected face; frames, total number of frames of the video; nb detect., number of detections; FP, observed rate of false positives for the face detector used; size, mean of detection width in pixels.

dataset because of the fast crossing and direction variation of pedestrians, and also the small size of detected faces (materials provided with this article contain these five videos). Video 6[a] is part of a dataset built to evaluate the detection of abandoned baggage systems in Advanced Video and Signal Based Surveillance Conference. The frame rate of videos Additional file 1: Video 1, Additional file 2: Video 2, Additional file 3: Video 3 and Video 6 is 25 frames per second and 13 frames per second for videos Additional file 4: Video 4 and Additional file 5: Video 5. To extract faces, we use the classic Viola–Jones [31] face detector implemented in OpenCV library. For all the videos, detections are rather small (from side 30 to 60 pixels) compared to the face recognition. It is close to the smallest face we can detect with the OpenCV implementation. Figure 5 shows the face images obtained from the face detector with our dataset.

The ground-truth clustering is made by hand by classifying all the detections, with a cluster for each people passage, plus a false positive cluster. Figure 6 shows some video screen-shots of our dataset.

7.2.2 Experimental procedures

In the following experiments, we compare the clustering stage in the MAP framework with two generic clustering methods using the same dissimilarity matrices. The first one is the hierarchical clustering with *single-link*, and the second one is a relatively new method based on affinity propagation between elements [22-24].

Then, we compare different movement dissimilarities: one based on optical flow, another with forward and backward tracking, and the last one with just the pixel distance between detections. We also present some results showing the impact of the skin color term P_f in detection appearance likelihood.

7.3 Results

7.3.1 Performance of the MAP clustering method

Table 2 compares the clustering stage of our MAP clustering method to hierarchical clustering and affinity propagation clustering. It shows the best F-measures observed for the three algorithms, using the same dissimilarity measure. For the movement dissimilarity, we use the one based on optical flow measure. Optimized F-measures are found by varying parameters which impacts the number of clusters. These parameters are dissimilarity cutting threshold for the hierarchical method, global preference parameter for affinity propagation, and P_e for our method.

Figure 5 Preview of face detections, we use the OpenCV implementation of the Viola–Jones face detector. Face images are obtained by growing the detection to the under-head area.

Figure 6 Preview of our dataset videos. From top left to bottom right: Additional file 1: Video 1, scene of videos Additional file 2: Video 2 and Additional file 3: Video 3, scene of videos Additional file 4: Video 4 and Additional file 5: Video 5, Video 6. Frame sizes in pixels: 800 × 600 for videos Additional file 1: Video 1, Additional file 2: Video 2 and Additional file 3: Video 3, 704 × 576 for 4–5, and 720 × 576 for 6 .

Results show that MAP clustering leads to better performances than the two other clustering methods. We can also see that hierarchical clustering outperforms affinity propagation. This is probably due to the fact that hierarchical clustering (with *single-link*) often suffers from the chain effect. In our case, the chain effect is not so problematic, mostly because one detection has high affinity with two other detections: one in previous frames the other in the next frames. This naturally gives clusters a chain shape. Affinity propagation selects exemplars in each cluster, so the clusters are grouped around exemplars. This cluster structure is not as suited to our application as the chain one.

The performance of the MAP clustering seems to come from the fact that it is mostly suited to the video-based detection situation. The two other methods just use dissimilarities and no prior on clusters shape, while the presented MAP model uses a first-order Markov chain to model clusters and handles false positives in a specific way.

7.3.2 Performance of our dissimilarity measure

Table 3 shows *F*-measures obtained with four versions of the presented algorithm. The first (*basic*) is the base algorithm without tracklets or optical flow, it just uses time,

Table 2 Best performances reached by three clustering algorithms with the same dissimilarity matrix

Videos	MAP	HAC	AP
1	**90.9**	88.9	*76.5*
2	**81.5**	73.4	*67.5*
3	**76.5**	67.3	*57.6*
4	**98.1**	88.8	*82.3*
5	**98.1**	92.3	*81.9*
6	77.7	**79.3**	54.0

MAP, presented method; HAC, hierarchical clustering, and AP, affinity propagation clustering. Best results are in bold and lowest in italic.

Table 3 *F*-measure (in %) of different methods and videos

Video	Basic	Tracklet	OF	No prior
1	80.4(6.9)	76.6(8.5)	**85.8**(**2.8**)	79.4(6.7)
2	70.8(6.5)	72.1(**2.2**)	**72.5**(5.6)	66.4(4.6)
3	**69.2**(2.9)	63.7(**2.5**)	68.7(**2.5**)	67.1(5.6)
4	87.8(6.7)	77.2(10.5)	**88.5**(7.2)	87.6(**6.6**)
5	94.8(2.3)	94.4(4.2)	**95.4**(**2.2**)	94.6(2.7)
6	75.8(5.9)	74.4(4.3)	**77.5**(6.0)	72.8(**3.5**)

Values represent mean and standard deviation of the *F*-measures for 100 values of the P_e parameter. Best results are in bold.

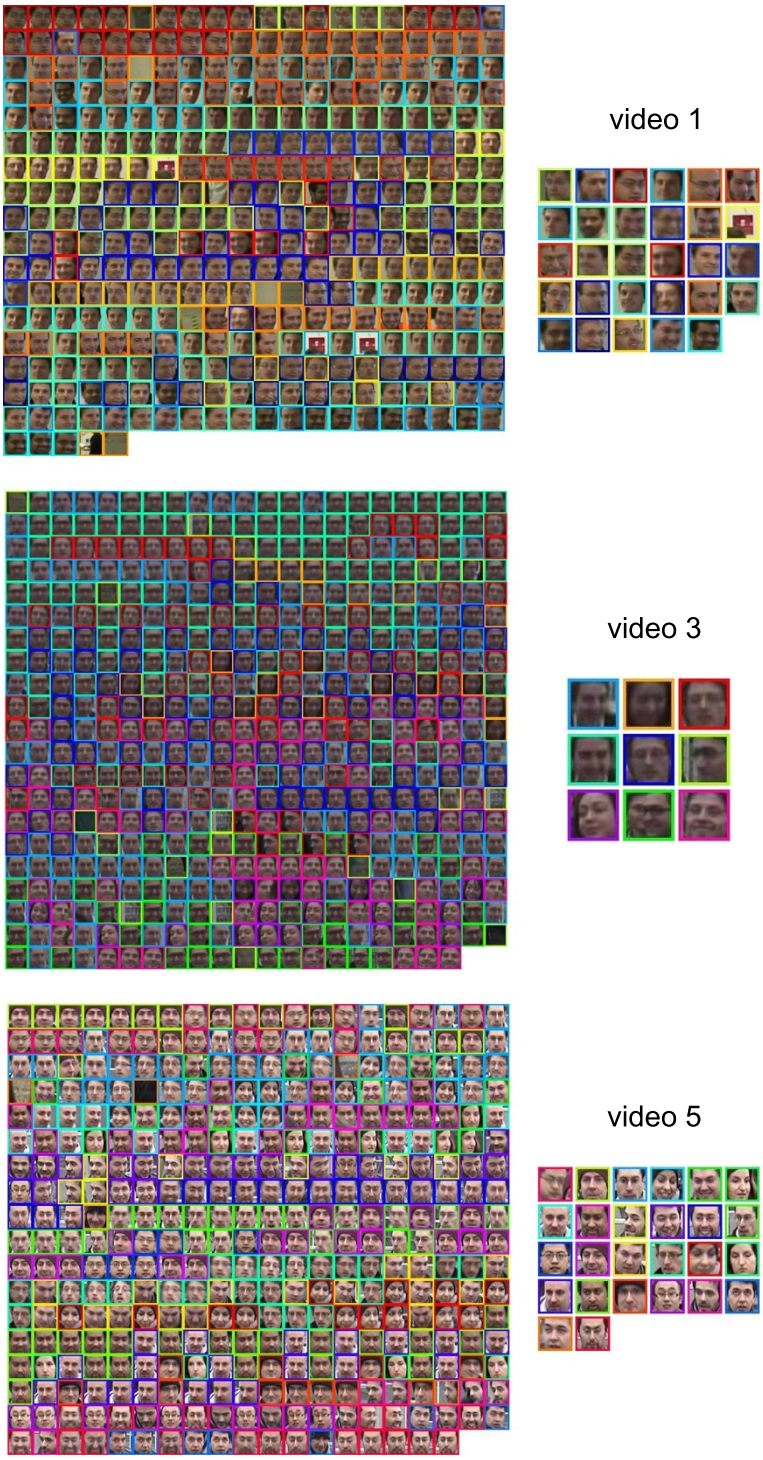

Figure 7 Example of face summarizations given by selecting a representative detection per cluster found, different colors represent different clusters. Left: 20% of the detected faces of a video (50% for Additional file 3: Video 3), right: selected faces.

appearance, and speed dissimilarities. The second version (*tracklet*) is based on the *basic* version but with the tracklet dissimilarities, it is the same for the third (*OF*) but with optical flow instead of tracklet dissimilarities. The last one (*no prior*) is like the *basic*, except it does not use the prior of the skin color ratio $P_f(\vec{a}_i)$ in the likelihood $P(\boldsymbol{z}_i|T)$.

The mean and standard deviation of F-measures are estimated over results obtained by varying the P_e parameter, which is the probability to start (or stop) a cluster at a

given detection. This parameter is used to force the number of clusters. For these experiments we take 100 values of P_e between 0.04 and 25%. Results show that the quality does not appear to be very sensitive to the large P_e variation.

Although dataset videos have various false positive rates (from 1.5 to 11%), we arbitrarily set the false positive parameter (β) to 1% for all the videos. This is done to avoid manual adjustment of *a priori* parameter to each video.

Table 3 shows that the use of optical flow dissimilarities gives better results compared to the use of dissimilarities based on tracklets. We can also see that the prior information based on skin color pixels improves results due to a better estimation of false positive cluster.

Concerning the running time, for Additional file 1: Video 1 (1934 frames and 1725 detections), the face detector takes 8 min 30 s to treat all the frames, the feature extraction stage (optical-flow version) takes 1 min 8 s, the computation of the dissimilarity matrix takes 8 s, and the optimization used to find the optimal clustering takes 20 s. We used a C++ implementation and run the test on a 2.4-GHz processor without parallelization. These figures give an overview of the different processing times, this indicates that most of the computation time is used in detection and image processing tasks.

As a qualitative and applicative result, we show (Figure 7) three examples of face summarization. After clustering face detections of a video, with MAP method (P_e fixed to 1%) and optical-flow dissimilarities, we remove small clusters and select a representative detection to build an album of faces. Detections are selected with a simple algorithm using a score based on detection size and contrast. As we do not use a re-identification process when people re-appear after being out of the field for a certain period, the main duplicates in faces summarizations are because persons cross several times the field of view.

7.4 Synthesis

In the main presented experiments, the method based on the MAP clustering is more suited to our problem than hierarchical clustering or affinity propagations are. The movement dissimilarity based on the optical flow improves the results, more than the tracklet dissimilarity does in the main cases. Using a skin color term in the face likelihood enhances clustering quality, by improving false positive cluster quality.

8 Conclusions

This article proposes a method to cluster face detections on challenging video sequences. Our method relies on a data-association framework by resolving a MAP problem. In the case of a frontal face detector, where detections are particularly sparse due to head rotations, experiments show that hat adding movement information to detection dissimilarities improves the results. Two different approaches are tested: the first based on short-term tracking and the second using optical flow extraction. We also present a new criteria to evaluate the performances of the resulting clustering.

Although our method has not reached the required quality for visual surveillance applications, we present a starting point for a video face summarization system based on tracking-by-detection, in scenes where automatic face recognition remains a challenging issue.

Consent

Consent was obtained from the persons appearing in the videos 1 to 5 used for this publication.

Endnote

[a]AVSS AB Hard from www.eecs.qmul.ac.uk/~andrea/avss2007_d.html.

Additional files

Additional file 1: Video 1.
Additional file 2: Video 2.
Additional file 3: Video 3.
Additional file 4: Video 4.
Additional file 5: Video 5.

Competing interests
The authors declare that they have no competing interests.

Acknowledgements
This study was supported by *Vesalis* and *ANRT* (France), we thank *Instut Pascal* and *Bio-Rafale* consortium. We also used the video AVSS AB Hard from the i-Lids dataset for AVSS 2007.

Received: 2 April 2012 Accepted: 23 November 2012
Published: 11 February 2013

References

1. D Ramanan, S Baker, S Kakade, in *11th IEEE International Conference on Computer Vision*. Leveraging archival video for building face datasets, (Rio de Janeiro, 14–21 Oct 2007). IEEE [http://www.ieee.org]
2. TL Berg, AC Berg, J Edwards, M Maire, R White, YW Teh, E Learned-Miller, DA Forsyth, in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2. Names and faces in the news. (Berkeley, 27 June-2 July 2004), pp. 848–854
3. M Everingham, J Sivic, A Zisserman, in *Elsevier Image and Vision Computing*, vol. 27. Taking the bite out of automated naming of characters in tv video (Elsevier, 2009), pp. 545–559. [http://www.elsevier.com]
4. J Sivic, M Everingham, A Zisserman, in *International Conference on Image and Video Retrieval*, vol. 3568. Person spotting: video shot retrieval for face sets, (Singapore, 2005), pp. 226–236
5. O Arandjelovic, A Zisserman, in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1. Automatic face recognition for film character retrieval in feature-length films, (Oxford, 2005), pp. 860–867
6. MC Nechyba, L Brandy, H Schneiderman, in *Multimodal Technologies for Perception of Humans: International Evaluation Workshops CLEAR 2007 and RT 2007*, vol. 4625. Pittpatt face detection and tracking for the clear 2007 evaluation, (2008), pp. 126–137
7. MC Nechyba, H Schneiderman, in *Multimodal Technologies for Perception of Humans: First International Evaluation Workshop on Classification of*

Events, Activities and Relationships, vol. 4122. Pittpatt face detection and tracking for the clear 2006 evaluation, (CLEAR, 2007), pp. 161–170. [http://www.clear-evaluation.org/]
8. M Zhao, J Yagnik, H Adam, D Bau, in *IEEE International Conference on Automatic Face and Gesture Recognition*. Large scale learning and recognition of faces in web videos. (Amsterdam, 17-19 Sept 2008)
9. M Kim, S Kumar, V Pavlovic, H Rowley, in *IEEE Conference on Computer Vision and Pattern Recognition*. Face tracking and recognition with visual constraints in real-world videos. (Anchorage AK, 23-28 June 2008)
10. F Bardet, T Chateau, D Ramadasan, in *12th IEEE International Conference on Computer Vision*. Illumination aware mcmc particle filter for long-term outdoor multi-object simultaneous tracking and classification, (Kyoto, 2009), pp. 1623–1630
11. S Dubuisson, J Fabrizio, in *Elsevier Pattern Recognition Letters* , vol. 30. Optimal recursive clustering of likelihood functions for multiple object tracking (Elsevier, 2009), pp. 606–614. [http://www.elsevier.com]
12. A Ess, B Leibe, K Schindler, L Van Gool, in *IEEE Conference on Computer Vision and Pattern Recognition*. A mobile vision system for robust multi-person tracking. (Anchorage AK, 23-28 June 2008)
13. B Leibe, K Schindler, L Van Gool, in *IEEE Proceedings of 11th International Conference on Computer Vision*. Coupled detection and trajectory estimation for multi-object tracking. (Zurich, 14-21 Oct 2007)
14. B Benfold, I Reid, in *IEEE Conference on Computer Vision and Pattern Recognition*. Stable multi-target tracking in real-time surveillance video. (Oxford, 20-25 June 2011),pp. 3457–3464
15. W Ge, R Collins, in *British Machine Vision Conference*, vol. 96. Multi-target data association by tracklets with unsupervised parameter estimation, (Leeds, 2008)
16. B Song, T-Y Jeng, E Staudt, AK Roy-Chowdhury, in *Springer 11th European Conference on Computer Vision*, vol. 6311. A stochastic graph evolution framework for robust multi-target tracking, (Heraklion, 2010), pp. 605–619
17. Q Yu, G Medioni, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31. Multiple-target tracking by spatiotemporal monte carlo markov chain data association (IEEE, 2009), pp. 2196–2210. [http://www.ieee.org]
18. J Berclaz, F Fleuret, P Fua, in *Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*. Multiple object tracking using flow linear programming, (Lausanne, June 2009)
19. C Huang, B Wu, R Nevatia, in *Proceedings of the 10th European Conference on Computer Vision: Part II*, vol. 5303. Robust object tracking by hierarchical association of detection responses, (Marseille, 2008), pp. 788–801
20. A Stergiou, G Karame, A Pnevmatikakis, L Polymenakos, in *Multimodal Technologies for Perception of Humans: International Evaluation Workshops CLEAR 2007 and RT 2007*, vol. 4625. The ait 2d face detection and tracking system for clear 2007, (Baltimore, 2008), pp. 113–125
21. L Zhang, Y Li, R Nevatia, in *IEEE Conference on Computer Vision and Pattern Recognition*. Global data association for multi-object tracking using network flows. (Los Angeles, 23-28 June 2008)
22. D Dueck, BJ Frey, in *IEEE International Conference on Computer Vision*. Non-metric affinity propagation for unsupervised image categorization. (Toronto, 14-21 Oct 2007)
23. BJ Frey, D Dueck, in *American Association for the Advancement of Science*, vol. 315. Clustering by passing messages between data points (Science, 2007), pp. 972–976. [http://www.sciencemag.org/]
24. Z Lu, MA Carreira-Perpinán, in *IEEE International Conference on Computer Vision*. Constrained spectral clustering through affinity propagation. (Portland, 23-28 June 2008)
25. T Brox, C Bregler, J Malik, in *IEEE Conference on Computer Vision and Pattern Recognition*. Large displacement optical flow. (Berkeley, 20-25 June 2009) pp. 41–48
26. J Marzat, Y Dumortier, A Ducrot, in *Proceedings of The 17th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG)*. Real-time dense and accurate parallel optical flow using cuda (WSCG, 2009). [http://www.wscg.eu/]
27. D Sun, EB Sudderth, MJ Black, in *Advances in Neural Information Processing Systems*, vol. 23. Layered image motion with explicit occlusions, temporal consistency, and depth ordering, (2010), pp. 2226–2234
28. N Rahman, K Wei, J See, in *Proceedings of The MMU International Symposium on Information and Communications Technologies*. Rgb-h-cbcr skin colour model for human face detection (Springer, 2006). [http://www.springer.com/]
29. P Perez, C Hue, J Vermaak, M Gangnet, in *7th European Conference on Computer Vision*, vol. 2350. Color-based probabilistic tracking, (Copenhagen, 2002), pp. 661–675
30. E Amigó, v Gonzalo, J Artiles, F Verdejo, in *Springer Information Retrieval*, vol. 12. A comparison of extrinsic clustering evaluation metrics based on formal constraints (Springer, 2009), pp. 461–486. [http://www.springer.com/]
31. P Viola, M Jones, in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1. Rapid object detection using a boosted cascade of simple features, (Cambridge, 2001), pp. 511–518

doi:10.1186/1687-5281-2013-10
Cite this article as: Schwab *et al.*: **A multi-cue spatio-temporal framework for automatic frontal face clustering in video sequences.** *EURASIP Journal on Image and Video Processing* 2013 **2013**:10.