Automatic Detection and Tracking of Pedestrians in Videos with Various Crowd Densities

Afshin Dehghan, Haroon Idrees, Amir Roshan Zamir, and Mubarak Shah

Abstract Manual analysis of pedestrians and crowds is often impractical for massive datasets of surveillance videos. Automatic tracking of humans is one of the essential abilities for computerized analysis of such videos. In this keynote paper, we present two state of the art methods for automatic pedestrian tracking in videos with low and high crowd density. For videos with low density, first we detect each person using a part-based human detector. Then, we employ a global data association method based on Generalized Graphs for tracking each individual in the whole video. In videos with high crowd-density, we track individuals using a scene structured force model and crowd flow modeling. Additionally, we present an alternative approach which utilizes contextual information without the need to learn the structure of the scene. Performed evaluations show the presented methods outperform the currently available algorithms on several benchmarks.

Keywords Human detection • Tracking • Data association • Crowd density • Crowd analysis • Automatic surveillance

1 Introduction

The number of surveillance cameras in urban area is increasing at a significant rate which results in massive amounts of videos to be analyzed. Observing crowds and pedestrians manually in such large amount of data is cumbersome and often impractical which makes automated methods extremely favorable for this purpose.

A. Dehghan • H. Idrees • A.R. Zamir • M. Shah (✉)
Computer Vision Lab, University of Central Florida, Orlando, USA
e-mail: adehghan@cs.ucf.edu; haroon@cs.ucf.edu; aroshan@cs.ucf.edu; shah@cs.ucf.edu

U. Weidmann et al. (eds.), *Pedestrian and Evacuation Dynamics 2012*,
DOI 10.1007/978-3-319-02447-9_1,

Automatic tracking of pedestrians is one of the required abilities for computerized analysis of such videos.

The density of pedestrians significantly impacts their appearance in a video. For instance, in the videos with high density of crowds, people often occlude each other and usually few parts of the body of each individual are visible. On the other hand, the full body or a significant portion of the body of each pedestrian is visible in videos with low crowd-density. These different appearance characteristics require tracking methods which suite the density of the crowd. In this paper, we present two state of the art methods for tracking pedestrians in videos with low and high density of crowds.

For videos with low density of pedestrians (Sect. 2), first we detect individuals in each video frame using a part-based human detector which efficiently handles occlusion (Sect. 2.1). Later, we employ a global data association method based on Generalized Minimum Clique Graphs for tracking each person over the course of the whole video (Sect. 2.2).

We present two approaches to tracking for videos with high density of crowds. In the first one, the scene layout constraint which is captured by learning Dynamic Floor Field, Static Floor Field and Boundary Floor Field along with crowd flow is leveraged to track individuals in the crowd. In the second approach, no learning or crowd flow is used to track targets. Instead, the tracking is performed utilizing salient and contextual information.

2 Pedestrian Tracking in Videos with Low Crowd Density

Our framework for tracking pedestrians in videos with low density of crowds consists of two main steps: Human Detection (Sect. 2.1) and Data Association (Sect. 2.2):

2.1 Part-based Human Detection

Human detection is a fundamental problem in video surveillance. Robust human tracking is highly dependent on reliable detection in each frame. Although human detection has been well studied in computer vision, most of the existing approaches are unsuitable for detecting targets with large variance in appearance. Therefore, robust human detection remains a challenge due to the highly articulated body postures, occlusion, background clutter and viewpoint changes.

Fig. 1 (**a**) A sample positive image and its HOG descriptor. (**b**) *Left*: detections obtained using part-based human detector in [6]. *Right*: a model for root and parts and a spatial model for the location of each part relative to the root

Many approaches have been proposed for human detection over the last decade. In most of them, the problem is formulated as a binary sliding window classification, i.e. an image pyramid is constructed and a fixed size window is scanned over all of its levels to localize humans using a non-maximum suppression procedure.

Dalal and Triggs [5] use HOG as low a level feature which is shown to outperform other competitive features, such as wavelets, for human detection. HOG provides a robust feature set that allows the human body to be distinguished discriminatively even in cluttered background. The descriptor purposed by Dalal and Triggs computes an edge oriented histogram on a dense grid of uniformly spaced cells. Then, they use overlapping local contrast normalizations in order to improve the overall performance. A linear SVM classifier is used to learn a model for the human body using positive and negative samples. The detector is then applied to the image to localize human bodies, i.e. the detector takes an image, a position within that image and a scale as the inputs and determines if there is a person in that particular location and scale. Figure 1a shows a sample positive image and its HOG descriptor.

Using local features to learn body parts is another approach to human detection. Part-based approaches which model an object as a rigid or deformable configuration of parts are shown to be very effective for occlusion handling. Felzenszwalb et al. [6] simultaneously learn parts and an object model. Their model is an enriched version of Dalal and Triggs' which uses a star structured part-based model defined by a root filter plus a set of parts associated using a deformation model. The score associated to each star model is the summation of the scores of the root filter and parts at a given location and scale minus a deformation cost which measures the deviation of parts from their ideal location relative to the root. The scores of both parts and root are defined as the dot product of a learnt filter which belongs to that part and a set of extracted features for that specific location. The same set of features as [5], i.e. HOG, is used in [6] with the difference that principle component analysis has been applied to HOG features in order to reduce the dimensionality.

Fig. 2 *Left*: human detection results using [6]. *Right*: human detection results using our approach where *red boxes* show the human detected as full bodies, *green boxes* show the humans detected as upper bodies, and *yellow boxes* show the humans detected as heads only. It is clear that [6] failed to detect occluded humans since it does not have an explicit occlusion model, while our approach detects the occluded parts and excludes them from the total detection scores, thus achieving significant improvements especially in crowded scenes

2.1.1 Human Detection with Occlusion Handling

While the deformable part-based model has recently shown excellent performance in object detection, it achieves limited success when the human is occluded. In particular, the final score in [6] is computed using the score of all the parts without considering that some of them can be occluded by other pedestrians or static objects in the scene. The occlusion happens especially in crowded scenes such as the example shown in Fig. 2 which signifies the drawback of this method. Considering the score of the occluded parts in the final decision score may cause the algorithm to ignore most of the partially occluded humans in the final detection results. Therefore, some methods such as [7] or [8] rely on head detection only and disregard the rest of the body.

To address this problem, we purpose in [9] to infer occlusion information from the score of the parts and utilize only the ones with high confidence in their emergence. By looking at the score of each part, we find the most reliable set of parts that maximizes the probability of detection. Let H denote the HOG feature of the image, and $p = (x,y)$ represent the location of a part. The detection score at location (x_0,y_0) defined in [6] is:

$$score\,(x_0, y_0) = b + \sum_{i=1}^{i=n} s\,(p_i)\,,$$

where b is the bias term, n is the number of parts, and $s(p_i)$ is the score of part i which is computed as:

$$s\,(p_i) = F_{p_i}.\varnothing\,(H, p_i) - d_{p_i}.\varnothing_d\,\left(d_x, d_y\right),$$

where F_{p_i} is the part filter, and $\varnothing\,(H,p_i)$ denotes the vector obtained by concatenating the feature vectors from H at the sub window of the part p_i (d_x,d_y) denotes the displacement of the parts with respect to the anchor position. To address the

discussed issue, instead of aggregating the score of all the parts, we select the subset of parts which maximize the detection score:

$$score\,(x_0, y_0) = b + argmax_{S_m} \frac{1}{|S_m|} \times \sum_{i \in S_m} \frac{1}{1 + exp\,(A\,(p_i)\,.s\,(p_i) + B\,(p_i))}\ .$$

The sigmoid function is introduced to normalize the score of the parts. The parameters A and B are learned by the sigmoid fitting approach and $|S_m|$ is the set cardinality. This equation corresponds to the average score of the parts in a subset which makes the comparison between different subsets easy.

If there is an occluded part in a subset $|S_m|$, its average score will be lower than a case which doesn't have any occluded parts. Therefore, by maximizing the above equation, we obtain the most reliable set of parts and its corresponding detection score. In our experiments we consider only three subsets of parts (full body, upper body and head only). We found these three subsets to be representative enough for most scenarios. That way, we do not need to search for all the 2^n parts. Figure 2 demonstrates the qualitative comparison between [6] and our approach.

2.2 *Data Association Using Generalized Graphs*

The method explained in Sect. 2.1 detects humans in each video frame. However, it does not specify which detections belong to one identity. We need to determine the detections which correspond to one particular pedestrian in order to form a trajectory. We employ a data association method based on Generalized Minimum Clique Problem (GMCP) for this purpose. The input to the data association method is the detections obtained using the human detector of Sect. 2.1, and the output is the trajectory of each pedestrian in the video. Figure 3 shows the block diagram of this process. First a video is divided into smaller segments and the human detector is applied to each video frame. Then, the GMCP-based data association method is utilized in order to form the tracklets of pedestrians in each segment. Later, we perform another data association using GMCP to merge the tracklets of one person found in different video segments into a full trajectory spanning over the course of the whole video.

2.2.1 Finding Tracklets of Pedestrians in One Video Segment

In order to determine if a group of detections from different video frames belong to one person, we utilize two features for each detection: Appearance and Spatial Location. If the visual appearances of a group of detections are similar and the tracklet they form is smooth, we conclude that they belong to one identity. On the other hand, if the appearances of some of the detections are not similar to the rest or if the trajectory they form includes abrupt jumps, we infer that some of the detections must belong to other pedestrians. In order to perform this task,

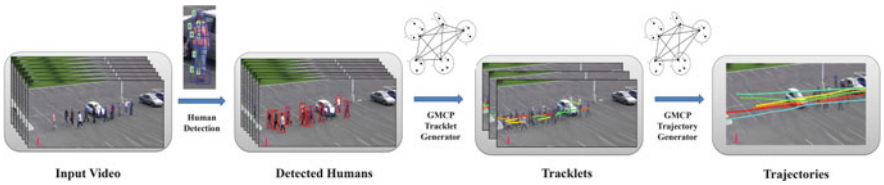

Fig. 3 Block diagram of the data association method. A video is divided into smaller segments, and the GMCP-based method is employed to find pedestrian tracklets and trajectories

we formulate the input to our data association problem as the graph $G = (\boldsymbol{V},\boldsymbol{E},\boldsymbol{W})$ where $\boldsymbol{V},\boldsymbol{E}$ and $\boldsymbol{W}$ denote the set of nodes, edges and edge weight respectively. Each node represents one human detection. The nodes in $\boldsymbol{V}$ are divided into a number of disjoint clusters. Each cluster represents one video frame and the nodes therein represent the detections in that particular frame. An edge weight is defined as the difference between the color histograms of two detections. Therefore, if two human detections are visually similar, the weight of the edge between their representing nodes is expected to be low and vice versa.

The solution to our data association problem is found by identifying one detection from each frame in a way that all the selected detections belong to one person. In other words, a feasible solution can be represented by a subset of the nodes of $\boldsymbol{G}$ which we call $\boldsymbol{V_s}$. We define the appearance cost of a feasible solution, $\gamma_{appearance}(\boldsymbol{V_s})$ as the summation of all the edge weights between its nodes. Therefore, by solving the optimization problem $\arg\min_{V_s}\left(\gamma_{appearance}\left(\boldsymbol{V}_s\right)\right)$, the feasible solution with the most consistent appearance features is found.

Generalized Minimum Clique Problem (GMCP) [11] is defined as selecting a subset of nodes from a superset in a way that the summation of edges weights between the selected nodes is minimized. The nodes in the superset are divided into a number of disjoint clusters. Exactly one node from each cluster should be included in the subset of selected nodes. As can be understood from the definition of GMCP, solving GMCP for the graph G is equivalent to solving our data association problem of $arg\min_{\boldsymbol{V_s}}\left(\gamma_{appearance}\left(\boldsymbol{V}_s\right)\right)$. Therefore, we find the generalized minimum clique of $\boldsymbol{G}$ in order to find the feasible solution $\boldsymbol{V_s}$ which has the minimum cost.

However, we add a term based on motion to our optimization function in order to incorporate the smoothness of trajectory in identifying the best feasible solution. Therefore, the optimal feasible solution, $\widehat{\boldsymbol{V}}_s$, is found by solving:

$$\widehat{\boldsymbol{V}}_s = arg\min_{\boldsymbol{V_s}}\left(\gamma_{appearance}\left(\boldsymbol{V}_s\right) + \gamma_{motion}\left(\boldsymbol{V}_s\right)\right).$$

The motion cost, $\gamma_{motion}(\boldsymbol{V_s})$, is based on the fact that humans tend to move smoothly and avoid unnecessary abrupt changes in direction and speed. Since a video segment usually covers a short temporal span of a few seconds, the motion of pedestrians therein can be assumed to be near constant velocity. We utilize a global motion model proposed in [10] in order to assign a cost to a feasible solution

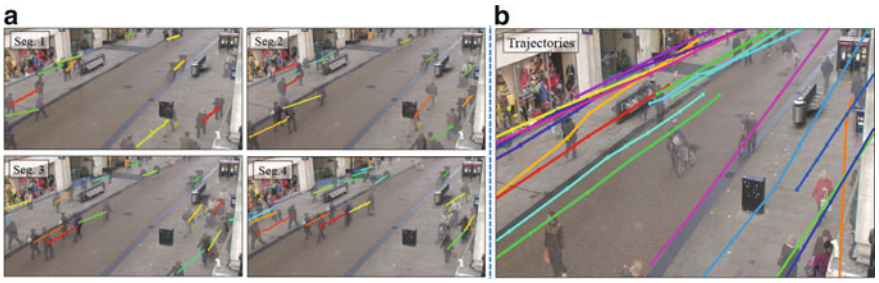

Fig. 4 (**a**) Tracklets found in four segments of a sample video sequence. (**b**) Tracklets are merged into full trajectories for all the pedestrians

based on motion. The employed motion model assigns a low cost to $\boldsymbol{V}_s$ if the corresponding tracklet follows constant velocity model and vice versa.

$\widehat{\boldsymbol{V}}_s$ found by solving the aforementioned optimization problem identifies the detections in different video frames which belong to one person. Therefore, by finding $\widehat{\boldsymbol{V}}_s$ the tracklet of one pedestrian in one video segment is found. Then, we exclude the nodes included in $\widehat{\boldsymbol{V}}_s$ from the graph $\boldsymbol{G}$ and solve the optimization problem again in order to compute the tracklet for the next pedestrian in the segment. This process continues until the time no or few nodes are left in graph $\boldsymbol{G}$ which implies all the pedestrians are tracked.

The human detector may fail to detect a pedestrian in one frame. This may happen due to several reasons such as occlusion, articulated pose or noise. Since GMCP selects one detection from each frame, it will choose an incorrect node for the frames where a particular person does not have a detection. Therefore, we add hypothetical nodes to each cluster which are supposed to represent virtual detections for the cases where human detector failed. The appearance features and spatial locations of hypothetical nodes are calculated based on the other detections included in $\boldsymbol{V}_s$ as explained in [10].

The tracklets found in four segments of a sample video sequence are shown in Fig. 4a.

2.2.2 Merging Tracklets into Trajectories

The explained process forms the tracklets of pedestrians in each video segment. In order to form the full trajectory of one person over the course of the whole video, we need to identify the tracklets which belong to one identity and merge them into a trajectory. This task in fact requires solving another data association problem. We employ a method similar to the one explained earlier in order to perform the association between tracklets. For this purpose, each tracklet in one segment is represented by one node. The appearance feature of each node is the average of color histograms of the detections in the corresponding tracklet. The spatial location of a node is defined as the middle point of the corresponding tracklet. We form an input graph similar to $\boldsymbol{G}$ and solve the optimization problem explained

Table 1 Tracking results on town center data set

	MOTA	MOTP	MODP	MODA
Benfold et al. [13]	64.9	80.4	80.5	64.8
Zhang et al. [14]	65.7	71.5	71.5	66.1
Pellegrini et al. [15]	63.4	70.7	70.8	64.1
Yamaguchi et al. [16]	63.3	70.9	71.1	64.0
Leal-Taixe et al. [17]	67.3	71.5	71.6	67.6
Ours/GMCP	**75.59**	**71.93**	**72.01**	**75.71**

Table 2 Tracking results on TUD and PETS 09 sequence

Dataset	MOTA	MOTP	Prec	Rec	IDsw
TUD-Crossing [18]	84.3	71.0	85.1	98.6	2
TUD-Crossing [19]	85.9	73.0	89.2	98.8	2
TUD-Crossing-Ours	**91.63**	**75.6**	**98.6**	**92.83**	**0**
TUD-Stadtmitte [20]	60.5	65.8	–	–	7
TUD-Stadtmitte-Ours	**77.7**	**63.4**	**95.6**	**81.4**	**0**
PET2009-View 1 [21]	80.00	58.00	81.00	60.00	28
PET2009-View 1 [22]	81.46	58.38	90.66	90.81	19
PET2009-View 1 [20]	81.84	73.93	96.28	85.13	15
PET2009-View 1 [23]	84.77	68.742	92.40	94.03	10
PET2009-View 1-Ours	**90.3**	**69.02**	**93.64**	**96.45**	**8**

earlier [11]. Doing that, the tracklets which include visually similar detections and form a smooth trajectory are associated together. Therefore, the trajectories of all pedestrians in the whole video are found. Sample result of the merging process is shown in Fig. 4b. The tracklets shown in Fig. 4a are merged to form the full trajectories shown in Fig. 4b.

2.2.3 Experimental Results

We evaluated the described data association method on four standard sequences. Town Center is a sequence of 4,500 frames which shows a semi-crowded scene. TUD-Crossing and TUD-Stadtmitte are two sequences with 201 and 170 frames respectively. PETS2009-S2L1 includes 800 frames with a challenging scenario because of frequent changes in the directions of the pedestrians.

Table 1 shows the tracking results for town center sequence along with comparison to the state of the art. MOTA and MOTP represent the accuracy and precision of tracking based on CLEAR MOT metrics [12]. Prec. and Rec. denote precision and recall value of assigning detections to their appropriate trajectories respectively. IDsw denotes number of ID-switches which represents the number of times a trajectory incorrectly switches between two different identities. Table 2 shows the evaluation results for TUD-Crossing, TUD-Stadtmitte and PET2009-S2L1 sequences. As can be seen, the presented data association method outperforms the state of the art on all the sequences.

The average time of performing data association is 4.4 s per frame using non-optimized Matlab code. The time complexity can be significantly improved upon availability of a parallel and optimized implementation in C.

3 Pedestrian Tracking in Videos with High Crowd Density

High density crowded scenes are characterized by a large number of individuals per unit area. With high density comes a new set of challenges that are not present in non-crowded scenes. These include a large number of individuals and their complex interactions, small target size, and difficulty in establishing correspondences due to proximity among individuals as well as occlusions caused by inter-object interactions. Furthermore, these challenges are dependent on density, the higher the crowd density, the more difficult it is to detect and track individuals. Figure 5 provides some examples of dense crowds.

3.1 Tracking in Dense Crowds Using Floor Fields

The first approach [25] we present for tracking high-density crowds leverages on the observation that the behavior of one individual in a crowded scene is dependent on its interactions with other individuals as well as structure of the scene. A model that captures these interactions in space-time can serve as an auxiliary source of information, thereby constraining the likely movement of individuals in the scene. Since movement of individuals in a crowd is restricted by other individuals and scene structure, we can treat the crowd as a collection of mutually interacting particles. At each point in the scene, we build a matrix of preferences that captures the likelihood of transition of a particle from one point in the scene to another point in its spatial neighborhood. Each transition is associated with a probability, where higher probability higher likelihood for a transition to occur. With inspiration from evacuation dynamics, where floor fields are manually specified for simulation purposes, in this approach, we automatically learn and model the interactions among individuals of a crowd through floor fields and use them for generating better predictions when establishing correspondences across frames (Fig. 6).

Static Floor Field (SFF) SFF captures the general movement of crowd in the scene, for instance, the dominant path taken by the crowd towards the preferred exit location. People tend to form crowds when they share the same goal and this goal-directed and rational behavior of crowds provides an important cue to the movement of individuals in the crowd. The process to compute SFF follows: First, optical flow is computed at each location for the initial Ns frames. The flow vectors are then averaged over Ns frames providing smoothed-out average flow at each location in the scene. Next, sink seeking process is performed to discover the sinks – attractive

Fig. 5 Examples of high-density crowded scenes

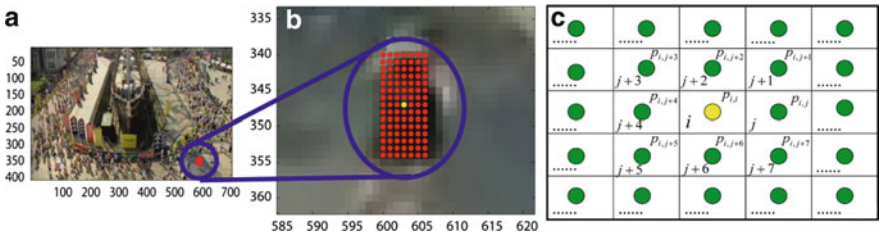

Fig. 6 In (**a**, **b**), the *red dots* are the particles on an individual while (**c**) shows the transition matrix that is obtained from the floor fields

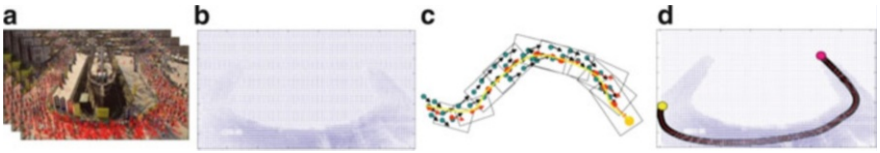

Fig. 7 Computation of Static Floor Field. (**a**) Shows the dense optical flow whereas (**b**) is the smoothed out flow. (**c**) Describes the sink-seeking process where *red dots* are velocity vectors from (**b**) for one particle, *cyan dots* are the neighbors, *orange dot* is the sink, whereas *rectangles* are the sliding windows. (**d**) Shows the path for one particle originating at *yellow* and ending at *red* (the sink)

regions towards which the individuals in the crowd move. For this, we initialize a grid of particles over computed flow field. Each particle moves under the influence of the flow field taking into account the influence from neighboring flow vectors (Fig. 7).

$$X_{i,t+1} = X_{i,t} + V_{i,t},\ V_{i,t} = \frac{\sum_{j \in neighbors} V_{j,t} W_{i,j,t}}{\sum_{j \in neighbors} W_{i,j,t}},\ W_{i,j,t} = \exp\left(-\left\| V_{i,t-1} - V_{j,t} \right\|^2\right)$$

where X is the location, V is the velocity, i denotes the individual and j is its neighbor. After performing sink seeking for each point in the scene, each point in a path is replaced by the number of steps required to reach the sink. Repeating this for all paths gives the SFF (shown in Fig. 8d).

Boundary Floor Field (BFF) BFF captures the influence from barriers and boundaries of the scene. Walls and boundaries tend to repel the individuals away from them. BFF is computed on NB frames from the future. First, crowd flow is

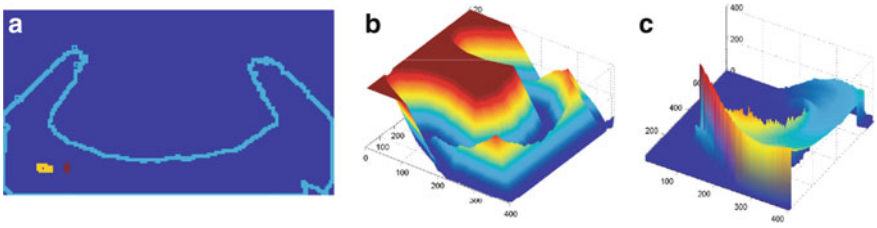

Fig. 8 (**a**) Is the FTLE field computed using [24] and (**b**) shows the boundaries computed as the derivative of (**a**). (**c**) Is BFF using distance transform on (**b**). (**d**) Is SFF obtained after sink-seeking process

segmented using the segmentation algorithm proposed in [24] where the boundaries in the flow field are computed as ridges in Finite Time Lyapunov Exponent (FTLE) field. The segmentation map is then used compute an edge map retaining only the boundary pixels. Next, for each point in the scene, its distance to the nearest barrier/boundary is computed using a distance transform thus, giving BFF. The larger the distance of a point from the boundary, the smaller its influence on an individual near that point. Figure 8a–c show the computation of BFF.

Dynamic Floor Field (DFF) DFF captures the instantaneous flow around point, using the ND frames in the future. The idea is similar to SFF but DFF is temporally localized compared to SFF. Stacking optical flow for ND frames into a 3D volume, a grid of particles is then overlaid and numerically advected while keeping counts of how many times a particle jumps from one point to another during advection. This gives a measure of dynamic interaction between points at each time instant, or DFF.

Tracking The probability for an individual at cell i transitioning to cell j is given by:

$$p_{ij} = Ce^{k_D Dij} e^{k_S Sij} e^{k_B Bij} R_{ij}$$

where Dij, Sij and Bij are the transition probabilities based on the three floor fields and kD, kS and,kB are the corresponding coefficients while Rij is probability based on appearance calculated using Normalized Cross Correlation.

Experiments We performed experiments on three marathon sequences for this approach. Sequence 1 has 492 frames and 199 individuals were selected for tracking, sequence 2 has 333 frames with 120 individuals, and 50 individuals were selected for tracking in sequence 3 which has 453 frames. Figure 9a–c shown the tracks obtained through the proposed approach. We compared this approach against MeanShift and the ground truth. Figure 9d shows a significant difference in tracking error between the proposed approach (green) and MeanShift (yellow). Figure 9e shows the comparison with the ground truth for selected individuals. The y-axis is the track length from proposed approach (black) and ground truth (green). As evident from the graph, for the 100 individuals compared, track length is very close to that of ground truth.

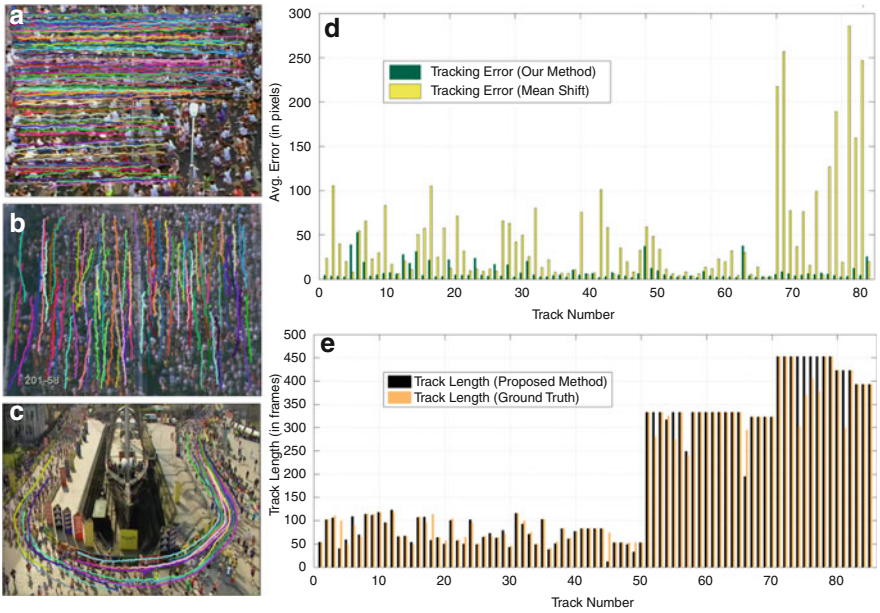

Fig. 9 (**a**–**c**) Tracked individuals in the three sequences. (**d**) Comparison with Meanshift. (**e**) Comparison with ground truth

3.2 Tracking in Dense Crowds Using Prominence and Neighborhood Motion Concurrence

The approach presented in previous section depends on learning the crowd flow, both averaged over time (SFF) as well as dynamic flow (DFF). The floor fields serve as a strong prior on the motion of individuals at each point in the scene. The assumption that an individual always behaves in a manner consistent with global crowd behavior does not always hold. The restriction on the motion of individuals from time-invariant priors may cause the tracker to fail when the crowd flow is dynamic, the crowd flow explores new region in the scene not previously learned, or when there is camera motion which may introduce errors in learning. In this section, we introduce the second approach [26] to the problem of tracking dense crowds in an online fashion without using any learning or crowd flow modeling.

Similar to the previous approach, we use Normalized Cross Correlation to obtain confidence for appearance. Owing to the challenges introduced by the high density crowds, the simplicity of template based tracker demands more than just appearance to perform well in crowded scenes. For that, we supplement the tracker with salient and contextual sources of information that significantly reduce the confusion in establishing correspondence across frames.

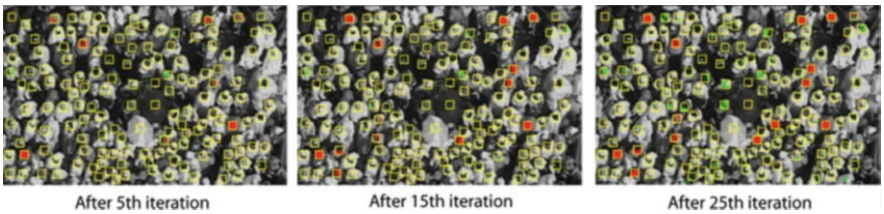

Fig. 10 Intermediate steps for the method of selecting prominent individuals. *Red* are the ground truth (manually selected) prominent individuals whereas *green* are the rest of the individuals. As evident, during back assignment, templates belonging to prominent individuals get filled first and therefore selected

Prominence The first idea in this approach is the prominence of individuals in terms of appearance. In any crowded scene, appearance of certain individuals will be different from the rest, that is, such prominent individuals can be tracked with high confidence.

In order to select prominent individuals, we generate features from the templates by extracting RGB values at each pixel. Then, we cluster all the features into k clusters using mixture of Gaussians. The clusters are sorted w.r.t density, where mass equals the number of points in that cluster and volume is given by $(2\pi)^{3/2}|\sum|^{1/2}$. Next, all the points in each cluster are assigned back to individual templates starting from the least dense cluster. This process of back assignment is stopped once T % of the templates are filled by at least two-thirds. Figure 10 shows the intermediate results of this procedure.

Neighborhood Motion Concurrence (NMC) The motion of an individual in dense crowd is similar to its neighbors. This information can be captured through a motion model that incorporates influence from neighbors. Let $x_i^t = [\mathrm{x}\ \dot{\mathrm{x}}]^T$ (position, velocity) , Σ_i^t represent the state and covariance of an individual i at time t, A be the 2×4 matrix that captures state transition, and $\aleph(\mu, \Sigma)$ a 2d Gaussian distribution. NMC for individual i with neighbors j is has two components, self and neighbor. The two components are given by:

$$p_S = p\left(z_i^{t-1} \middle| \widehat{x}_i^{t-1}\right) . \aleph\left(A x_i^{t-1}, A \Sigma_i^{t-1} A^T\right),$$

$$p_N = \sum_j w_j . \aleph\left(A x_{ij}^{t-1}, A \Sigma_j^{t-1} A^T\right),$$

$$w_j = \frac{\exp\left(-\left\|\mathrm{x}_j - \mathrm{x}_i\right\|\right)}{\sum_{k \in Neighbors} \exp\left(-\left\|\mathrm{x}_k - \mathrm{x}_i\right\|\right)}.$$

Figure 11b, c is an illustration depicting NMC. Black Gaussian in Fig. 11 © corresponds to self-component of the individual under consideration (black square in Fig. 11b) while the colored Gaussians show the neighbor component of NMC.

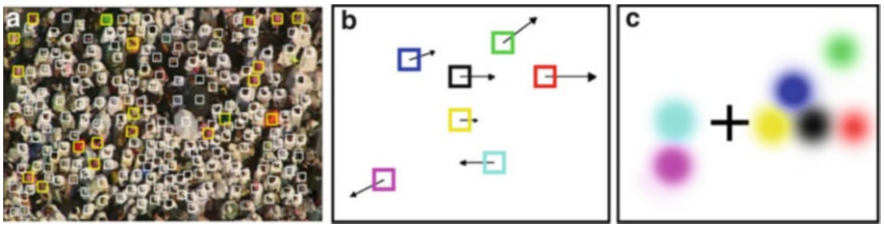

Fig. 11 Final output of the procedure to detect prominent individuals. (**b**) *Black square* is the individual under consideration while *colored squares* are its neighbors. *Black arrows* show the velocities with which the individuals are moving. (**c**) *Cross hair marks* the position of individual with *black square* in (**b**). *Colored Gaussians* are the corresponding contributions to NMC of individual under consideration

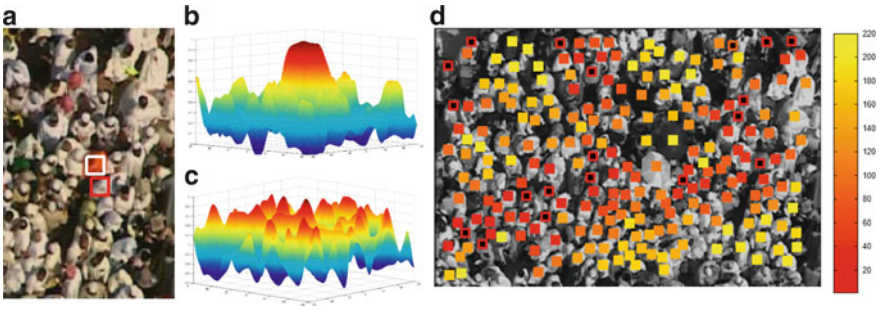

Fig. 12 Hierarchical update. (**a**) *White square* marks a prominent individual whereas *red square marks* its non-prominent neighbor. (**b**, **c**) Are their appearance surfaces which shows that prominent individuals with their unique appearance are less likely to be confused with the neighbors and therefore should be places at the top of tracking hierarchy. (**d**) Shows the order in which individuals in this scene were updated (*red* to *yellow*)

Hierarchical Update After having defined the notions of prominence and NMC, the final aspect of this approach is the order in which individuals in a crowded scene are updated. In Fig. 12, the prominent individual (white square in Fig. 12a) has appearance surface given in Fig. 12b, whereas its neighbor (red square in Fig. 12a) has the appearance surface given in Fig. 8c. It is evident that prominent individuals have less confusion with their neighbors and they should be placed on top of the tracking hierarchy. The algorithm, therefore, starts by updating prominent individuals, followed by their neighbors and continues till all the position of all individuals is updated. The position of non-prominent individuals is updated using NMC.

Results We compared approach presented in this section with MeanShift, Normalized Cross Correlation tracker, MeanShift Belief Propagation as well as the approach based on floor fields presented in previous section. The experiments were performed on nine sequences of medium to high density. In Table 3, various characteristics of the nine sequences such as number of frames, number individuals

Table 3 Quantitative comparison for the two approaches against Normalized Cross Correlation, MeanShift and MeanShift Belief Propagation for nine crowded sequences of medium to high density

	Seq 1	Seq 2	Seq 3	Seq 4	Seq 5	Seq 6	Seq 7	Seq 8	Seq 9
# Frames	840	134	144	492	464	333	494	126	249
# People	152	235	175	747	171	600	73	58	57
Template size	14	16	14	16	8	10	10	10	14
NCC	49 %	85 %	57 %	52 %	33 %	52 %	50 %	86 %	35 %
MeanShift	19 %	67 %	17 %	8 %	7 %	36 %	28 %	43 %	11 %
MSBP	57 %	97 %	80 %	69 %	62 %	81 %	68 %	94 %	45 %
Floor fields	75 %	99 %	85 %	84 %	66 %	92 %	67 %	97 %	57 %
Prominence/NMC	80 %	100 %	93 %	94 %	72 %	94 %	67 %	92 %	63 %

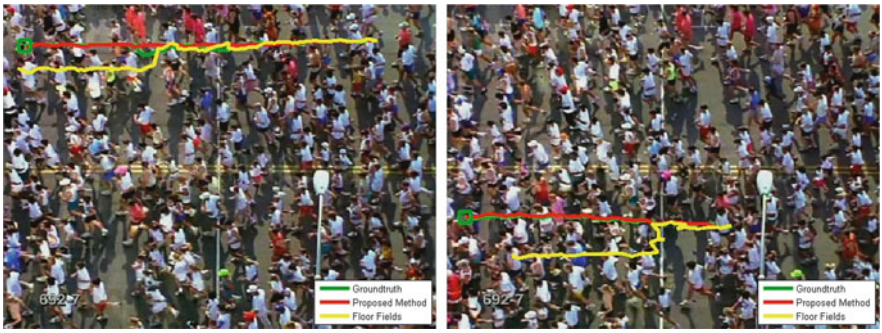

Fig. 13 This figure provides the comparison between the two approaches presented. *Green* is the ground truth trajectory, *yellow* is from first approach and *red* is from the second approach

and template size used for tracking are given in first three rows. Tracking accuracy is reported for the five methods and nine sequences as a percentage of number of points (in all trajectories) that lie within 15 pixel threshold.

Qualitative comparison between the two approaches is given in Fig. 13. The three cases are presented where the second approach (red) performs better than floor fields (yellow). Ground truth trajectory is drawn in green.

4 Conclusion

Automatic tracking of pedestrians is essential for computerized analysis of surveillance videos. In this keynote paper, we present two state of the art tracking methods for videos with low and high density of crowds. The method for low density scenarios first detects pedestrians in each video frame using a part-based human detector. Then, a data association method based on Generalized Graphs is employed for tracking each individual in the whole video. For videos with high crowd density, two approaches are presented. The first one is based on using the scene structure

based force model, while the second approach utilizes contextual information. Our Experiments show the presented frameworks outperform the currently available methods on several benchmarks.

References

1. E. Osuna, R. Freund, and F. Girosi, "Training Support Vector Machines: An Application to Face Detection," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 130–136, 1997.
2. C. Papageorgiou, T. Evgeniou, and T. Poggio, "A Trainable Pedestrian Detection System," Proc. Symp. Intelligent Vehicles, pp. 241–246, 1998.
3. P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," Proc. IEEE Conf. Computer Vision and Pattern Recognition, vol. 1, pp. 511–518, 2001.
4. Y. Wu, T. Yu, and G. Hua, "A Statistical Field Model for Pedestrian Detection," Proc. IEEE Conf. Computer Vision and Pattern Recognition, vol. 1, pp. 1023–1030, 2005.
5. N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," Proc. IEEE Conf. Computer Vision and Pattern Recognition, vol. 1, pp. 886–893, 2005.
6. P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. In PAMI, 2010.
7. B. Benfold and I. Reid. Stable multi-target tracking in realtime surveillance video. In CVPR, 2011.
8. M. Rodriguez, I. Laptev, J. Sivic, and J.-Y. Audibert. Density-aware person detection and tracking in crowds. In ICCV, 2011.
9. G. Shu, A. Dehghan, O. Oreifej, E. Hand, M. Shah. Part-based Multiple-Person Tracking with Partial Occlusion Handling. In CVPR, 2012.
10. Amir Roshan Zamir, Afshin Dehghan, and M. Shah. GMCP-Tracker: Global Multi-object Tracking Using Generalized Minimum Clique Graphs. In ECCV, 2012.
11. Feremans, C., Labbe, M., Laporte, G.. Generalized network design problems. In: EJOR, 2003.
12. Kasturi, R., et al.: Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. In: PAMI. (2009).
13. Benfold, B., Reid, I.: Stable multi-target tracking in real time surveillance video. In: CVPR.(2011)
14. Zhang, L., Li, Y., Nevatia, R.: Global data association for multi-object tracking using networkflows. In: CVPR. (2008)
15. Pellegrini, S., Ess, A., Van Gool, L.: Improving data association by joint modeling of pedestrian trajectories and groupings. In: ECCV. (2010)
16. Yamaguchi, K., Berg, A., Ortiz, L., Berg, T.: who are you with and where are you going? In: CVPR. (2011)
17. Leal-Taixe, L., Pons-Moll, G., Rosenhahn, B.: Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. In: ICCV Workshops. (2011)
18. Breitenstein, M.D., Reichlin, F., Leibe, B., Koller-Meier, E., Gool, L.V.: Robust tracking-by-detection using a detector confidence particle filter. In: ICCV. (2009)
19. Brendel, W., Amer, M., Todorovic, S.: Multiobject tracking as maximum weight independent set. In: CVPR. (2011)
20. Andriyenko, A., Schindler, K.: Multi-target tracking by continuous energy minimization. In:CVPR. (2011)
21. Berclaz, J., Fleuret, F., Turetken, E., Fua, P.: Multiple object tracking using k-shortest paths optimization. In: PAMI. (2011)

22. Shitrit, H.B., Berclaz, J., Fleuret, F., Fua, P.: Tracking multiple people under global appearance constraints. In: ICCV. (2011)
23. Henriques, J.F., Caseiro, R., Batista, J.: Globally optimal solution to multi-object tracking with merged measurements. In: ICCV. (2011)
24. S. Ali and M. Shah, A Lagrangian Particle Dynamics Approach for Crowd Flow Segmentation and Stability Analysis, IEEE CVPR, 2007.
25. S. Ali and M. Shah, Floor Fields for Tracking in High Density Crowded Scenes, ECCV 2008.
26. H.Idrees, N. Warner and M.Shah, Tracking in Dense Crowds using Prominence and Neighborhood Motion Concurrence, Submitted to CVIU Journal, 2012.