Image and Vision Computing 32 (2014) 14–26

Contents lists available at ScienceDirect

Image and Vision Computing

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

Tracking in dense crowds using prominence and neighborhood motion concurrence ☆

Haroon Idrees [a,*], Nolan Warner [b,1], Mubarak Shah [a]

[a] Center for Research in Computer Vision (CRCV), University of Central Florida, Orlando, United States
[b] Department of Computer Science and Engineering, University of Nevada, Reno, United States

ARTICLE INFO

Article history:
Received 24 January 2013
Received in revised form 25 September 2013
Accepted 30 October 2013

Keywords:
Crowd analysis
Dense crowds
Tracking
Prominence
Neighborhood motion concurrence
Hierarchical tracking

ABSTRACT

Methods designed for tracking in dense crowds typically employ prior knowledge to make this difficult problem tractable. In this paper, we show that it is possible to handle this problem, without any priors, by utilizing the visual and contextual information already available in such scenes.

We propose a novel tracking method tailored to dense crowds which provides an alternative and complementary approach to methods that require modeling of crowd flow and, simultaneously, is less likely to fail in the case of dynamic crowd flows and anomalies by minimally relying on previous frames. Our method begins with the automatic identification of prominent individuals from the crowd that are easy to track. Then, we use Neighborhood Motion Concurrence to model the behavior of individuals in a dense crowd, this predicts the position of an individual based on the motion of its neighbors. When the individual moves with the crowd flow, we use Neighborhood Motion Concurrence to predict motion while leveraging five-frame instantaneous flow in case of dynamically changing flow and anomalies. All these aspects are then embedded in a framework which imposes hierarchy on the order in which positions of individuals are updated. Experiments on a number of sequences show that the proposed solution can track individuals in dense crowds without requiring any pre-processing, making it a suitable online tracking algorithm for dense crowds.

1. Introduction

Crowd analysis is an active area of research in Computer Vision [1]. Over the past few years, methods have been proposed that estimate density and number of people in a crowd [2,3], find group structures within a crowd [4], detect abnormalities [5–8], find flow segments [9,10], and track individuals in a crowd [11–13].

Density is an important feature which can be used to classify different kinds of crowds [1]. From the computer vision perspective, videos of high density crowds can be divided into groups based on the number of pixels on target. High density crowds with extremely small object size permit only holistic approaches for scene understanding, such as finding motion patterns and segmentation of crowd flows [14,15,9,10]. However, if individuals in a crowd are distinguishable, then tracking of individuals may be possible, which is important in the context of safety and surveillance [1].

Tracking in dense crowds [16,12,13] is a challenging problem. Non-crowd methods, which track individuals in isolation do not perform well on dense crowds [16] because the large number of objects in close proximity poses difficulty in establishing correspondences across frames. Furthermore, human motion in crowds is influenced by social constraints [17] which force individuals to follow more complex, non-linear patterns that need to be taken into account for successful tracking of dense crowds.

Methods specifically designed for dense crowds generally require some learning of motion priors, which are later employed for tracking. For instance, Ali and Shah [16] proposed an algorithm which is based on the assumption that all individuals in a crowd behave in a manner consistent with global crowd behavior and learn the direction of motion at each location in the scene. The floor fields they learn severely restrict the permitted motion that individuals in a particular scene can have. This restriction on the motion of individuals due to time-invariant priors would cause the tracker to fail when, (1) the crowd flow is dynamic, (2) the crowd flow shifts or moves to a new region which was not learned before, and (3) when there are anomalies. Furthermore, camera motion and jitter can make learning the crowd flow difficult, if not impossible. Though learning, whether online or offline, certainly helps in tracking dense crowds when these issues are not present, our goal in this paper is to emphasize the use of visual and contextual information available in such

☆ This paper has been recommended for acceptance by Xiaogang Wang.

* Corresponding author.
E-mail address: haroon@eecs.ucf.edu (H. Idrees).

[1] This work was initiated when the author visited University of Central Florida, Orlando, for the National Science Foundation's REU program.

0262-8856/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.imavis.2013.10.006

crowded scenes to track in an online manner, without any pre-processing, learning or crowd flow modeling.

At the core of our approach lies template-based tracking, which is used to obtain the probability of observation. However, the simplicity of a template-based tracker demands more than just appearance to perform well in high density crowds. We supplement the tracker with novel visual and contextual sources of information, which are particularly relevant to crowds and reduce the confusion in establishing correspondences.

The first idea we explore is prominence of individuals which is similar to saliency (generally used for features and points). In any crowded scene with a large number of people, the appearance of some individuals will be markedly different from the rest (Fig. 1). The probability of confusing such individuals with the rest of the crowd will be low. Thus, the prominence of such individuals provides extra information which should be leveraged in tracking.

The second idea is to employ influence from neighbors to make better prediction for an individual's position. This idea is based on the observation that individuals in dense crowd experience social forces that bound their movement. For instance, an individual cannot jump across its neighbors in a single time instance. The restriction on movement that each individual experiences is proportional to the density of the crowd. Social force models, both in computer graphics and vision, are generally geared towards collision avoidance, where the goal is to predict positions such that subjects or individuals don't collide with each other. Our model, on the other hand, exploits the fact that movement of individuals in a dense crowd is similar to their neighbors, and therefore can be used to make better predictions.

Combining prominence and influence from neighbors, our method imposes an order on the way positions of individuals are updated. Individuals with prominent appearance are updated first, which subsequently guide the motion of the rest of the crowd. While updating, if the underlying patch-based tracker gives weak measurement for an individual, then position of the individual is updated based on appearance-based dense instantaneous flow. Thus, the framework we introduce incorporates these ideas as well as their inter-relationships. Our contributions in this paper can be summarized as,

- An alternative approach to dense crowd tracking which highlights the significance of prominence and spatial context for tracking dense crowds without requiring crowd flow modeling,
- Introduction of the notion of prominent individuals, its relevance to tracking in dense crowds, and a method to detect prominent individuals,
- Incorporation of influence from neighbors, prominent or not, to better predict and estimate an individual's position,
- A tracking framework which imposes an order in the way individuals are tracked, where positions of prominent individuals are updated first and individuals with low probability of observation from underlying tracker are updated last.

Fig. 1. An example of a dense crowd where individuals that are in yellow squares stand out from the crowd and, therefore, should be easier to track than rest of the individuals, marked with white squares.

Since space is complementary to time, both the visual information (prominence) and spatial context (influence from neighbors) are complementary to temporal constraints (crowd flow, motion patterns) introduced in previous works on tracking people in dense crowds. Our goal in this work is to emphasize the first two, which when coupled together allow tracking in an online fashion, without modeling crowd flow and without looking at observations from the future.

2. Related work

There are a few papers that have used context for tracking, however, there is currently no such work that utilizes it for dense crowds. Yang et al. [18,19] used contextual information to improve the tracking performance of a few objects. Through color segmentation of the image, they find auxiliary objects, which are easier to track and whose motion is correlated with the target. The auxiliary objects are then tracked; they also aid in tracking the target, which occurs simultaneously. The method was streamlined for non-crowd scenarios, with results containing a maximum of three objects per sequence. Furthermore, due to hundreds of people frequently occupying the entire screen in crowd videos, the definition and discovery of auxiliary objects is not applicable to crowd sequences. Khan et al. [20] also capture interaction between targets using particle filters in an MRF framework. However, they do not consider prominence and anomalies while tracking, and the particle filters are not suitable for crowd sequences due to fewer pixels per target.

The methods proposed for multi-target tracking include Park et al. [21] who sped up belief propagation using mean shift by sparsely sampling the belief surface instead of using parametric methods or non-parametric methods that require dense sampling. They do not assume prominence and pass messages in all directions, therefore presuming absence of anomalies.

Next, we review papers relevant to different aspects of our method. For an in-depth analysis of crowd literature, interested readers are referred to the survey by Zhan et al. [1].

2.1. Prominence

Discriminative features were used for tracking by Collins and Liu [22], who rank the foreground features online and track objects using only those features which discriminate foreground from background. A similar idea was explored by Mahadevan and Vasconcelos [23] who, given a pool of features from foreground and background, select the most informative features for classification between the two. In relation to our method, prominence can be seen as a *collection* of salient features which discriminates one foreground object from the rest.

2.2. Social force models

Static motion models (such as linear velocity or constant acceleration) have long been used for tracking in computer vision. Dynamic models, as opposed to static ones, account for the dynamic structure of the scene and objects, and are based on the fact that individuals are driven by goals and respond to changes in their environments by adjusting their paths. Methods that model [24–26] and simulate crowds [27] incorporate this crucial information to produce realistic results.

In computer vision, social force models have been used for multi-target tracking, such as Pellegrini et al. [28]. They introduced Linear Trajectory Avoidance, a model inspired by Helbing and Molnár [17], in which predictions are made so that individuals avoid collisions with each other and the obstacles. The repulsive forces are balanced by a preference of each individual to move towards a destination with some desired speed, both of which are assumed to be known in advance. The experiments were performed on non-crowded scenes, since collision avoidance has lower applicability to dense crowds where individuals have less freedom of movement. To overcome some of the shortcomings in [28], Yamaguchi et al. [29] proposed a similar approach using a more sophisticated model, which tries to predict destinations and groups among individuals using certain heuristics based on trajectory features and classifiers trained on annotated sequences. They use very simple scenes and assume people move along straight paths, with a time-invariant notion of destination. There are scenes in real world where this assumption will break, for instance, Sequence 5 in Section 4.

Furthermore, Yamaguchi et al. [29] penalize deviations from preferred speed, which is set to 1.3 m/s. This is the speed at which an average human walks, but this constant will be different for a scene depicting a marathon, where people can be seen running at various speeds. In fact, both the methods [28,29] assume that the positions of individuals are in metric space, where distances can be computed between individuals in terms of metric units (meters). This is a natural disadvantage of sophisticated social force models whose parameters, otherwise, would have to be learned for each testing video anew. Secondly, for correct transformation of positions of individuals from image space to metric space, both methods assume that the video be captured from a bird's-eye view even if there are only a few individuals at any given time. These two strict assumptions limit the applicability of their methods to arbitrary videos. Although some camera elevation is necessary to completely capture a dense crowd, our model which works in the image space can work with slightly slanted views, i.e., lower than a bird's-eye view, because we anchor motion of all individuals on prominent ones who lie in the same scene as the rest of the crowd. In other words, since we impose motion consistency in the image space, we do not require knowing the transformation between image and metric coordinates, as such transformations cannot be assumed to be known in advance for arbitrary videos.

2.3. Dense crowds

Recently, Garg et al. [30] addressed the problem of matching instances of people in images of crowded events using photographs from Flickr. Unlike our problem, which deals with single-view videos, their method works on images taken from the same scene which allow structure from motion and 3D reasoning to match the subjects.

For tracking in dense crowds, in a series of papers, Kratz and Nishino [31–33] trained Hidden Markov Models to learn motion patterns in the scene which they later use for tracking individuals. Our method provides an alternative to such training-based methods by using appearance and contextual information only. The method proposed by Song et al. [13] tracks individuals by learning patterns of flow through online clustering of tracked trajectories. Wu et al. [12] did not learn any priors but employed multiple cameras to obtain 3D trajectories of objects that are indistinguishable in terms of appearance by finding correspondences across the multiple views.

The work most similar to ours is that of Ali and Shah [16] where authors use transition probabilities computed from learned floor fields in order to track individuals in a dense crowd. The method requires a pre-processing period where the static floor field is learnt using particles advected through optical flow across the scene. Furthermore, the dynamic floor field which captures the instantaneous flow is a non-causal process as it uses observations from the future. Similarly, Rodriguez et al. [34] use Correlated Topic Model (CTM) to capture different overlapping and non-overlapping crowd behaviors in the scene. In their construction, words correspond to low level quantized motion features and topics correspond to crowd behaviors. Similar to [16], the method requires temporal modeling of crowd behavior which uses observations from the future.

Recently, Rodriguez et al. [35] also proposed a method that solves the same problem, but instead of learning crowd flow, they build a database of approximately five hundred videos and match patches from query videos to the database videos. Their method requires extensive searching of similar patches in the database, while making a strong assumption that the motion of individuals in a particular query patch can be found in the database. We, on the other hand, rely completely on information that is readily available in the sequences.

Therefore, our goal is to develop an online tracker for dense crowds without requiring extensive analysis of sequences in the database, or off-line analysis by modeling the crowd behavior in advance. Instead, we explore visual and spatial information in this work in the form of prominence and influence from neighbors while making sure that the method is not biased against anomalies or dynamic crowd flow like the previous methods. Since temporal information is complementary to spatial and visual constraints, the proposed method can be seen as an alternative and complementary approach to previous methods for tracking individuals in structured dense crowds.

Furthermore, due to difficulty of human detection in dense crowds, and to keep the primary focus on tracking, all previous works in this area [16,34,35,33] assume that a manual initialization of templates on individuals in the crowd is afforded to the algorithm. The template refers to a bounding box around the individual that we intend to track. In this paper, like previous works, we also assume that initial templates (bounding boxes) are provided and our goal is to track them across the scene. This also restricts the applicability of other social force [36] or tracking methods which perform data-association among human detections across frames of the video.

3. Framework

The proposed framework augments template-based tracker, which alone yields poor results due to extreme difficulty in establishing correspondences in a densely crowded scene (See Section 4). In this section, we first discuss the notion of prominent individuals and present an algorithm that identifies such individuals. Then, we present the Neighborhood Motion Concurrence model which gives a probability surface of position for an individual using position and velocity information of the target and its neighbors. Finally, we develop a tracking method which updates the position of individuals in an ordered fashion using information about prominent individuals, influence from neighbors, and feedback from template-based tracker.

3.1. Prominence

Although it is possible to track and update the positions of all individuals in a crowd simultaneously at each time step, this is not the most efficient method. Some individuals have unique characteristics that make them stand out from the crowd. These characteristics make it easier to establish correspondences across frames for these individuals without confusing them with the rest of the crowd. The first step, therefore, would be to detect prominent individuals, whom we will refer to as *Queen Bees* or, in short, queens. We choose to use this term because a queen, due to its size, is the only unique bee in an entire colony of indistinguishable bees. Since a queen is unique and easily

Algorithm 1 Algorithm to find queens given templates $T_{1:n}$

procedure DETECTQUEENS
  $\phi = [\,]$ ▷ feature matrix
  for all $i = 1$ to n **do**
    $\phi_i =$ features from T_i ▷ feature =: an RGB vector per pixel
    Concatenate ϕ_i to ϕ
    $\Omega(\phi_i) = i$ ▷map from features to templates
    $N_i = |\phi_i|$ ▷ | | =: cardinality
  end for
  $[C, \mu, \Sigma] =$ GMM (ϕ, k) ▷ k =# of clusters
  Sort clusters w.r.t density i.e. $|C|/(2\pi)^{3/2}|\Sigma|^{1/2}$
  $V_{1:n} = [0\ 0 \ldots 0]$ ▷ 1xn voting array
  $queens = [\,], i = 0$
  while $i \leq k \wedge |queens| < .1n$ **do**
    $i = i + 1$
    for all $\phi \in C^i$ **do** ▷ $\phi \subseteq \phi$, C^i =: ith cluster
      $V_{\Omega(\phi)} = V_{\Omega(\phi)} + w(\phi, i)$ ▷ w= voting function
    end for
    $queens = \{j \mid V_j > \frac{2}{3} N_j\}$
  end while
end procedure

identifiable, it can be used to describe prominent targets in any type of dense crowd.

There are several features that can make an individual prominent with respect to others, such as gait [11], physical appearance, height or age. However, in dense crowds with a small number of pixels on a target, visual appearance is robust and, typically, the only observable feature. In our framework, a queen is defined as an individual with color features that differ significantly from the majority of the crowd.

To select the queens, we extract features ϕ_i from the templates $T_{1:n}$. Every pixel in each template gives one 3D feature, i.e., [*R*,*G*,*B*] at that pixel. While generating the features, we keep a map, Ω, that associates features to the source templates. All the features are then clustered into k clusters modeled using a mixture-of-Gaussians distribution, i.e., each component is $\mathcal{N}(\mu, \Sigma)$. Next, the clusters $C^{1:k}$ are sorted in ascending order according to a particular criterion (density). Finally, the features are reassigned to their original (source) templates beginning with features from the first cluster in the sorted list. The process is stopped once a small percentage of total templates (in our case, 10%) are filled by at least two-thirds. Since all the features from each cluster are processed simultaneously, it is possible to have more than 10% of total templates selected as queens. Algorithm 1 gives a general and formal description of this procedure.

For a cluster C, its mass m is given by $|C|$ (i.e., the number of data points in C), and volume v given by $(2\pi)^{3/2}|\Sigma|^{1/2}$. Then there are several ways to sort clusters: mass (m), volume (v), mass weighted by volume ($m{\cdot}v$), density (m/v) or the reciprocal of density (v/m). We ran a small experiment to find which criterion gives the best results for prominence by determining if it correctly identifies the queens while filling few non-queen templates (red and green in Fig. 3, respectively). We ran the experiment several times to avoid differences due to clustering, and

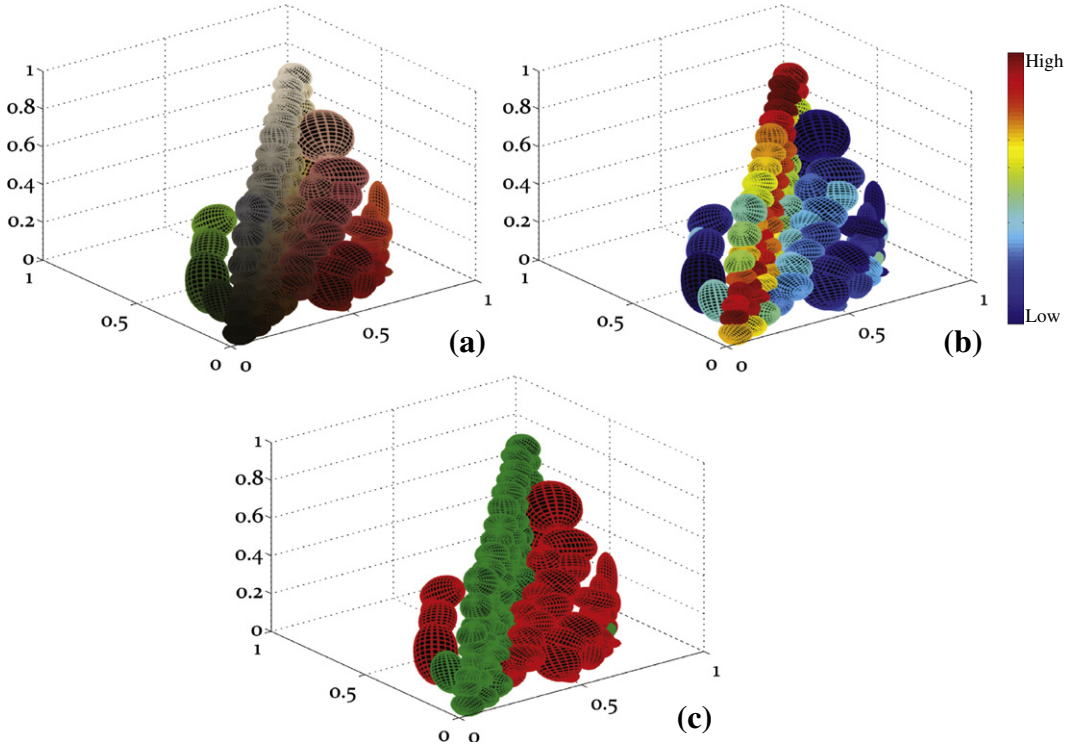

Fig. 2. Visualization of clusters: Given a fixed set of templates, we extract [*R*,*G*,*B*] features for each pixel in the template. We keep a map ($\boldsymbol{\Omega}$) between features and templates, i.e., we associate the id of each template with its constituent features. Then, the features are clustered and modeled using a Mixture-of-Gaussians distribution. The results on the image and templates in Fig. 1 are shown in (a) where each Gaussian is represented with an ellipsoid drawn with size equal to 1.5 the size of variance, i.e., 1.5 $(\boldsymbol{\Sigma})^{1/2}$ and colored with its mean, i.e., $\boldsymbol{\mu}$. The colors belonging to non-queens (white templates in Fig. 1) form clusters along the diagonal (black to white). In (b) we color the ellipsoid according to the density of the respective Gaussians with sparse clusters in blue and dense clusters in red. (c) The clusters that were used in selecting the queens are given in red after which the process of back-assignment stopped and did not use clusters drawn in green.

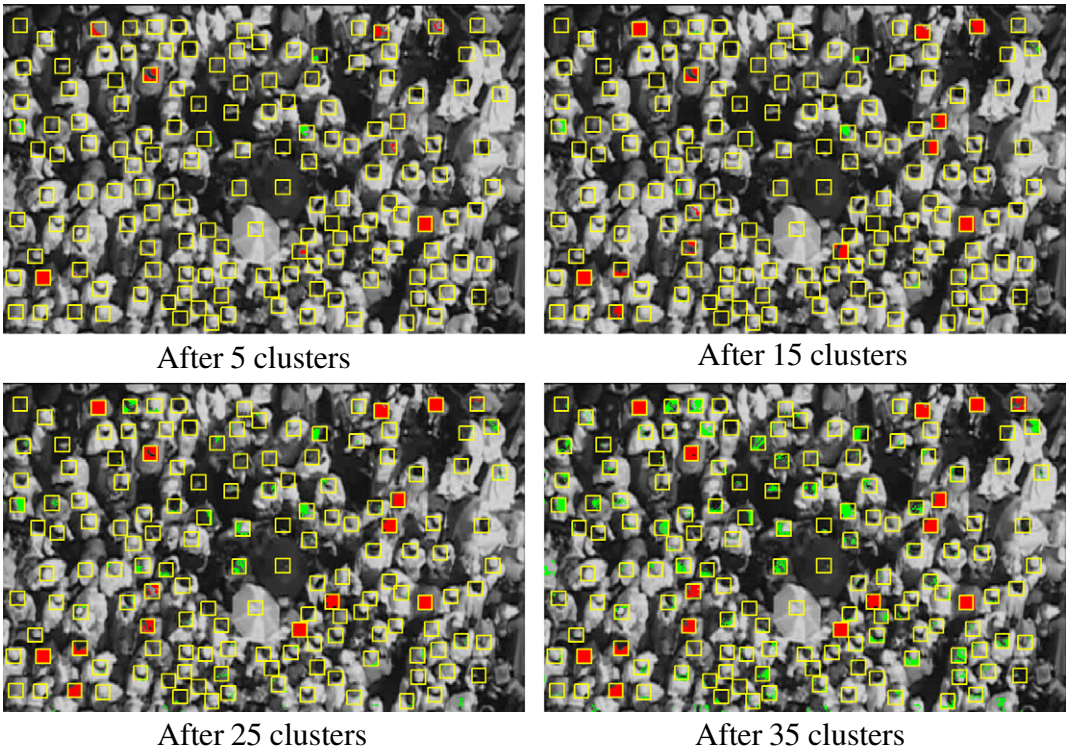

Fig. 3. Intermediate outputs for the queen detection method: The images correspond to back-assignment after processing $k = 5$, 15, 25 and 35 clusters (out of $k = 100$). Red and green colors indicate queens and non-queens, respectively. Notice that the proportion of green regions to red increases as the number of clusters increases.

found that density gives the correct and most stable queens across iterations. This implies that features of queens behave as outliers during clustering and form sparse clusters, whereas features of non-queens form dense clusters since they tend to be similar to each other.

For the image given in Fig. 1, the results of clustering are shown in Fig. 2(a). In this figure, each cluster is drawn in the RGB space using an ellipsoid whose size and orientation are determined by Σ and the color is given by μ. In Fig. 2(b), we color-code the clusters according to density where blue indicates sparse clusters and red indicates dense clusters. Fig. 2(c) shows the clusters whose features were utilized during back-assignment. These clusters are shown in red and the features in these clusters primarily belong to queens, whereas the features belonging to clusters in green were not used because the desired number of queens had already been identified through back-assignment by that time. The intermediate results of back-assignment for the image in Fig. 1 are shown in Fig. 3 where the procedure stopped after processing 35 clusters. The final results are shown in Fig. 1 where the yellow templates mark the selected queens while white templates belong to non-queens.

3.2. Neighborhood Motion Concurrence (NMC)

In this section, we present an intuitive model, that utilizes the dynamic contextual information of the crowded scene, which allows us to track individuals in a dense crowd without requiring any prior knowledge (crowd flows, motion patterns, …). Let $x_i^t = [\mathrm{x}\ \dot{\mathrm{x}}]^T$ (position, velocity), Σ_i^t represent the state and covariance, respectively, of target i at time t, $\hat{x}_i^t$ be the updated state, A be the state transition matrix, for instance, linear velocity and $\mathcal{N}(\mu, \Sigma)$ a 2D Gaussian distribution. We will distinguish the target under consideration from its neighbors by using subscripts i and j, respectively.

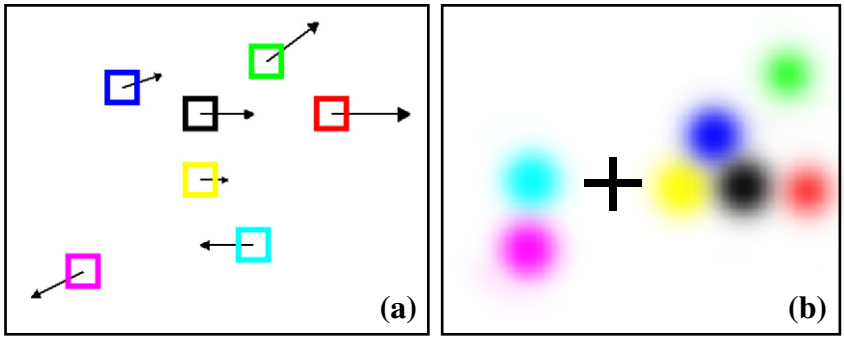

Fig. 4. Visualization for Neighborhood Motion Concurrence (NMC) model: (a) The target under consideration whose position is to be updated is shown with black square while its updated neighbors are shown with different colors. The arrows show the velocity which, for the target is velocity at t-1, whereas, for neighbors is their velocity at t. (b) shows the probability of position using the model for the target in (a), given by Eq. (5). The cross hair represents position of the target before the update, i.e., position at t-1. Each blurred circle represents $\mathcal{N}(\mu, \Sigma)$. The black circle is obtained using constant velocity assumption on the motion of target (p_S from Eq. (1)), while colored circles capture the influence from neighbors (p_N from Eq. (2)), based on their respective motions. This is a simple illustration, so each covariance is assumed to be an identity matrix.

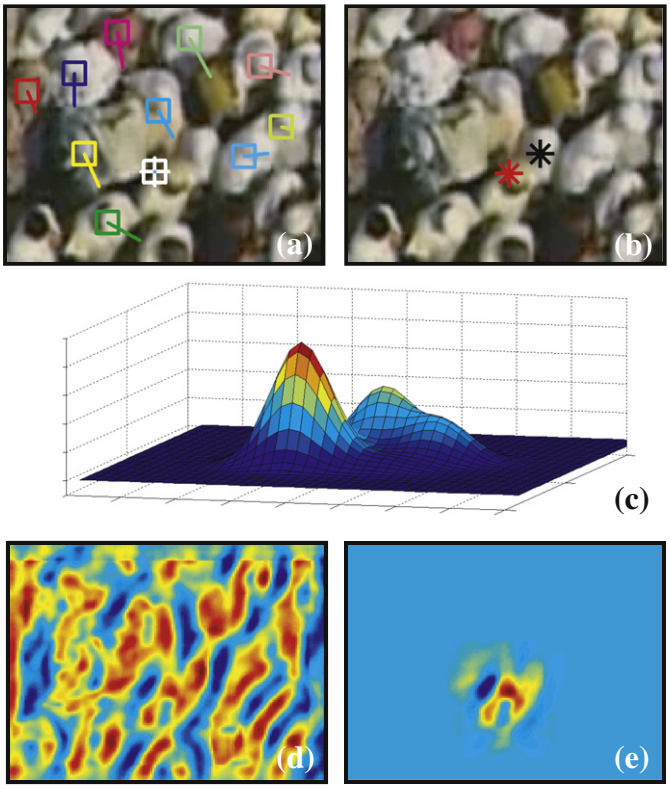

Fig. 5. (a) The target under consideration is shown in white and its updated neighbors in color. (b) The red star is the correct position updated by the method, whereas black star is the incorrect position update from template-based tracker alone. Intermediate results: (c) is the probability surface of position using NMC for the target in (a). The bottom row shows the effects of using the model, where (d) and (e) are probability surfaces of position with and without the model, respectively.

The Neighborhood Motion Concurrence has two components, namely self, p_S, and neighbors' influence, p_N. Since the state of the target under consideration at time t has not been updated yet, both its position and velocity come from the previous time instant *t-1*,

$$p_S = p\left(z_i^{t-1}|\hat{x}_i^{t-1}\right)\cdot\mathcal{N}\left(Ax_i^{t-1}, A\Sigma_i^{t-1}A^T\right), \tag{1}$$

where $p(z_i^{t-1}|x_i^{t-1})$ denotes the observation likelihood z_i^{t-1} of the target given its state x_i^{t-1}, obtained through underlying tracker. If there was some uncertainty in the target's position at time *t-1*, then p_S gets weighed down by this factor, therefore more preference will be given to prediction from neighbors, which is the second component of NMC, given by,

$$p_N = \sum_j w_j\cdot\mathcal{N}\left(Ax_{ij}^t, A\Sigma_j^t A^T\right)\cdot\lambda_j, \tag{2}$$

where $x_{ij}^t = \left[x_i^{t-1}\ \dot{x}_j^t\right]$, $\lambda_j = 1$ if the state of target j has been updated before i at time t, and 0 otherwise. But, not all influences can be treated equally, so we weigh them according to the neighbors' distance from the target,

$$w_j = \frac{exp\left(-\left\|x_j - x_i\right\|\right)}{\sum_{k\in Neighbors} exp(-\|x_k - x_i\|)}. \tag{3}$$

To illustrate the idea, consider the target shown in black square in Fig. 4(a) whose position is to be updated at time t, i.e., the black square is drawn where the target was at time *t-1*. Its updated neighbors are shown with squares of different colors, whose positions are depicted at time t. The arrows originating from the center of the squares indicate the velocity of each individual, i.e., velocity at time *t-1* for target and at time t for the updated neighbors. In Fig. 4(b), we show how each velocity vector from Fig. 4(a) influences the likelihood of the target's position. In this image, the cross-hair marks the position of the target before it is updated. The blurred circles represent Normal distributions ($\mathcal{N}(\mu, \Sigma)$). The black circle represents p_S, while colored circles represent p_N, using the same colors as the squares depicting neighbors in Fig. 4(a). Here, all covariances are set to identity matrices for the sake of visualization. Thus, NMC generates a probability distribution which gives a dynamic prior on the motion of the target based on its own motion and that of its neighbors. Fig. 5(a) shows a real example of the use of this model. The position of an individual in white square with cross-hair is to be updated, while some of its neighbors have already been updated, shown in colored squares. The lines originating from the center of the squares show the velocity vectors. Fig. 5(c) is the output of the model, which is a multi-modal distribution with one strong peak. Fig. 5(d) shows the probability of the target's position using just the appearance, while Fig. 5(e) shows the drastic reduction in confusion in the target's position achieved with the model, which is typical to majority of the non-queen individuals of the crowd. The final results are shown in Fig. 5(b) where the black star represents the incorrect position updated without using NMC, and the red star indicates the correct position updated with NMC.

3.3. Appearance based instantaneous flow

The Neighborhood Motion Concurrence models the similarity of motion of individuals in a dense crowd. The assumption that an individual has motion similar to its neighbors is violated when there are multiple flows in close vicinity of each other, for instance, two groups of people walking right next to each other in opposite directions, or when there are anomalous individuals in the crowd whose motion is not consistent with their neighbors. In these cases, we resort to instantaneous flow (Fig. 6), which provides some information about the possible direction of motion for such individuals. We construct instantaneous flow from five frames using normalized cross-correlation on patches that are densely initialized throughout the scene, where track of each patch captures temporally-localized motion. The idea is similar to particle advection [9], however when the duration is only five frames, particle advection gives results significantly worse than instantaneous flow due to noisy and inconsistent optical flow. We initialize 4×4 patches at a regular spacing of 4 pixels. When NMC does not provide good prediction, e.g., when the observation likelihood at the updated position is low, we approximate the motion using nearby patches in an instantaneous flow field. In this case, the neighborhood component p_N changes and $x_{ij}^t = \left[x_i^{t-1}\ \dot{y}_j^t\right]$, where $\dot{y}_j^t$ is the velocity of the patch averaged over the five frames. The selection of neighbors in Eq. (3) is now based on nearby patches instead of individuals.

In the next section, we use these three aspects together, which allow us to formulate a solution to the challenging problem of tracking in dense crowds without relying on any prior knowledge.

3.4. Tracking in dense crowds

Given some initialization, our goal is to track each individual in the crowd. If the crowd flow has been modeled in advance, then it is possible to update the positions of all individuals simultaneously. However, a sequential approach is preferable when the flow is not known. For each individual at each time instant, a decision needs to be made for the position update, which allows us to assign a confidence to this decision. Thus, tracking can be posited as a decision making process where

queens serve as guides and NMC is used for consensus. This idea lends itself to a hierarchical framework which starts with queens and ends with non-queens.

At the top of the tracking hierarchy are queens, which are updated first. Fig. 7 justifies their placement at the top of hierarchy. The two targets, one queen and an adjacent non-queen are shown with red and white squares, respectively. In Fig. 7(b), we show the probability surface of position using just the appearance for the queen, while in Fig. 7(c), we show the same for the non-queen. The surface in Fig. 7(c) is common to non-queens which signifies greater possibility of confusion among them, in this case, due to the white appearance of nearly all the neighbors. It is evident that the queen's neighbor in Fig. 7(c) will pose a significant challenge in tracking unless its state predictions are guided by the adjacent queen.

Once the queens are updated, their immediate neighbors are updated next, then the neighbors of neighbors. This process continues to expand outward until every target has been updated at the current time. If $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denotes the graph where $\mathcal{V}$ is the targets represented by their states and $\mathcal{E}$ is the edges between each individual and its neighbors within a fixed radius, then the updating order is Breadth First Search which can be implemented using a queue.

Let $NN_i \equiv \{j | e_{ij} \in \mathcal{E}\}$ be the neighbors for target i. For simplicity of notation, index j will represent a member of NN_i. Given states and covariances of a target i and its neighbors j, under first-order Markov assumption,

$$p\left(x_i^t | z_i^t, x_i^{t-1}, x_j^t\right) \propto p\left(z_i^t | x_i^t\right) p\left(x_i^t | x_i^{t-1}, x_j^t\right). \tag{4}$$

The state of queens is updated first, which is predicted by their own previous state, i.e., $p(x_i^t | x_i^{t-1}, x_j^t) = p_S$. However, the neighbors of a non-queen target whose state has been updated at time t influence its state estimate using NMC, given by,

$$p\left(x_i^t \,|\, x_i^{t-1}, x_j^t\right) = \zeta(p_S + p_N), \tag{5}$$

where ζ is the normalization factor.

In Eq. (4), $p(z^t|x^t)$ is the probability of measurement given state at time t, which corresponds to confidence from the tracker. Our underlying tracker is template-based, with Normalized cross-correlation as the similarity measure, hence,

$$p\left(z^t \,|\, x^t\right) = \frac{1}{2}\left(\gamma\left(\mathrm{x}^t\right) + 1\right), \tag{6}$$

where $\mathrm{x} = (u,v)$ and $\gamma(\mathrm{x})$ is given by

$$\frac{\sum_{x,y,z} \overline{f}(x,y,z) \cdot \overline{T}(u-x, v-y, w-z)}{\sqrt{\left(\sum_{x,y,z} \overline{f}(x,y,z)\right)^2 \left(\sum_{x,y,z} \overline{T}(u-x, v-y, w-z)\right)^2}}, \tag{7}$$

where z iterates over color channels and $\overline{T}$ is the target template T with its mean subtracted.

Finally, the state with maximum posterior probability is given by,

$$\hat{x}_i^t = \underset{x_i^t}{\arg\max}\ \ p\left(x_i^t | z_i^t, x_i^{t-1}, x_j^t\right). \tag{8}$$

Note that the distribution in Eq. (5) depends only on the neighbors whose position has been updated. If $p\left(z^t | \hat{x}^t\right)$ for a target is low, which might be due to poor prior probability from NMC or occlusion, we then resort to instantaneous flow for obtaining prior probability. The prior probability for such individuals is based on neighboring patches in the instantaneous flow field which is similar to Eq. (2) except that x_{ij}^t now comes from patches rather than individuals. However, if the update confidence does not improve when using Instantaneous Flow, we

Algorithm 2 Algorithm to update state given templates: $T_{1:n}$, NN graph: $\mathcal{G} = (V, \mathcal{E})$, state vectors at time $t-1$: $X_{1:n}^{t-1}$, and id of queens $Q = \{q \mid q \subset \{1, 2, \ldots n\}\}$

procedure HIERARCHICALUPDATE
  $\forall i \mid i \in \{1, 2, \ldots, n\}, \lambda_i = 0, \eta_i = 0$ ▷ λ: if updated, η: # visits
  Insert $\{q \mid q \in Q\}$ into queue
  while do $\exists j \mid \lambda_j = 0$
    Retrieve element i from start of queue

    Generate $p(x_i^t \mid x_i^{t-1}, x_j^t)$ from NMC
    Generate $p(z_i^t \mid x_i^t)$ according to Equation 6
    Find $\hat{x}_{i,NMC}^t$ by Equation 8
    if $p(z_i^t \mid \hat{x}_{i,NMC}^t) < \tau$ **then**
      Generate $p(\hat{x}_i^t \mid x_i^{t-1}, x_j^t)$ based on instantaneous flow
      Find $\hat{x}_{i,IF}^t$ by Equation 8
    end if

    if $p(z^t \mid \hat{x}_{i,NMC}^t)^{\frac{1}{\eta_i+1}} > \tau \vee p(z^t \mid \hat{x}_{i,IF}^t) > \tau$ **then**
      $\lambda_i = 1$
      Push $\{j \mid j \in NN_i \wedge \lambda_j = 0\}$ into queue
      in order of increasing distance
    else
      $\eta_i = \eta_i + 1$
      if $i \in Q$ **then**
        $Q = Q \backslash \{i\}$
      end if
      Push i at the end of queue
    end if
  end while
end procedure

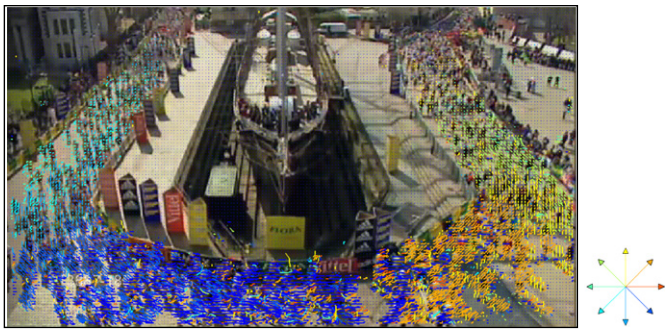

Fig. 6. This figure shows instantaneous flow computed for one of the frames of Sequence 5. The patches were densely initialized at a spacing of 4 pixels. The direction at each location in the image is shown with color wheel on bottom-right of the image.

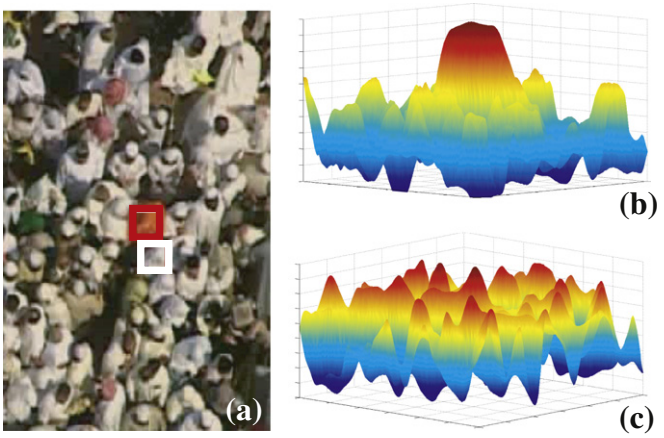

Fig. 7. (a) The red square marks one of the queens and the white one is its immediate non-queen neighbor. (b) shows the probability surface (obtained through Eq. (6)) of the queen's position in the next frame while (c) is the corresponding probability surface for its neighbor. It is obvious that queens, due to a uni-modal probability distribution, have less confusion in maintaining identity than non-queens, and therefore should be placed at the top of tracking hierarchy.

delay the update of such target and place it at the end of the queue so that it does not influence the rest of the crowd. Placing the target back into queue when tracker confidence is low has the peril of running into an infinite loop. The theorem (see Appendix A) shows that it is not possible in case of Algorithm 2 which, in effect, gradually lowers the threshold till the individual's position is updated.

Fig. 8(a) shows the results of hierarchical update for one of the frames in Sequence 2 containing 220 individuals. The order of update is color-coded with bar shown on the right, where red signifies individuals whose state was updated before the ones shown in yellow. The queens are marked with a black square inscribed in a red square. In some instances, yellow squares occur in close proximity highlighting the delayed update where we wait till more neighbors get updated, or update such individuals based on instantaneous flow (Algorithm 2, Line 15). In Fig. 8(b), we show the final graph produced by the update scheme. The arrows indicate the direction in which the influenced was transmitted. An interesting observation regarding Algorithm 2 is that initially, when updating queens, we do not use any information from neighbors. However, as we move down the hierarchy and away from the queens, we begin to employ more information from neighbors. In this figure, state prediction for several non-queen targets (orange to yellow) is influenced by neighbors, which were adjacent to different queens. Therefore, as we move away from the queens, the confidence due to prominence subsides, however, it is somewhat compensated by information from an increased number of updated neighbors down the hierarchy.

3.4.1. Relationship to Bayesian Networks

The hierarchical order for tracking we propose in this work is similar to belief propagation on a graph with directed edges but no cycles, which is equivalent to directed acyclic graph (DAG), or a Bayesian Network (Fig. 8(b)). The evidence is provided by the underlying tracker, and our goal is to find an estimate for the positions of all individuals in the crowd given their respective evidence. The conditional probabilities are mixture-of-Gaussians distributions and are provided by NMC. The update scheme starts with the prominent individuals, followed by their neighbors in a one-by-one fashion. Since edges only emanate from nodes (individuals) whose states (positions) have already been updated, the topology of the network evolves and changes till states of all individuals in the scene have been updated. Hierarchical update can, thus, be seen as a single pass of messages over this time-varying Bayesian Network. An additional advantage of this updating scheme is that it allows handling of anomalous motions, e.g., individuals whose motion does not conform with their neighbors. Since, if the probability of state given local evidence is low for an individual, we ignore the messages received from other individuals and resort to instantaneous flow, which would not have been possible if we used a simultaneous solution. Fig. 9 provides two instances from Sequence 5 where the proposed method was successfully able to track anomalous individuals whose movement significantly deviated from the rest of the crowd.

4. Experiments

We tested the proposed method on a variety of sequences which differed in terms of crowd density and tracking difficulty. There are a total of 8 sequences depicting commuters walking outdoors (Sequence 1–2: high density), marathons with people running at various speeds (Sequence 3–6: high density), and railway stations (sequence 7: medium density and 8: low density). The first frame of each sequence is shown in Fig. 10 in the first and third columns, with the corresponding sequence number at the bottom-right of the frame. We manually annotated the eight sequences with the total number of individuals annotated in each sequence ranging from 58–747. Some statistics on these sequences are shown in the first three rows of Table 1.

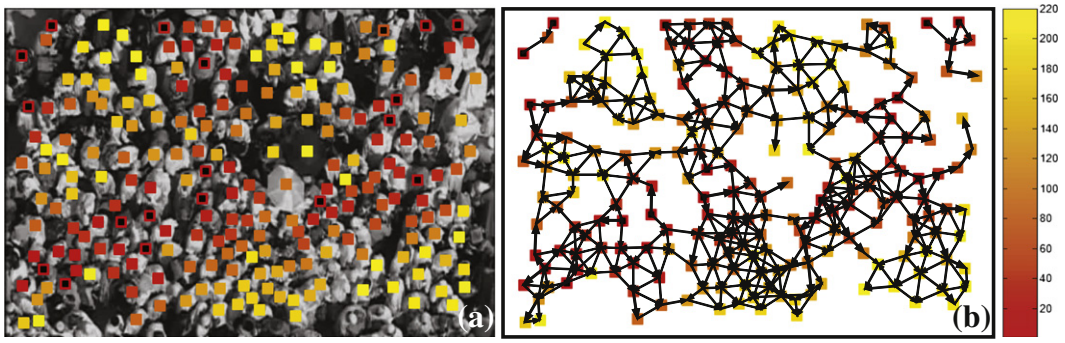

Fig. 8. Hierarchical update: (a) This image shows the order of update from Algorithm 2 on Sequence 2 which contained 220 people. The colors are encoded with the color-bar on the right, with red indicating individuals that were updated earlier than the ones shown in yellow. The update scheme starts with queens (black square inscribed in a red square), and moves down the *hierarchy* till all of the individuals are updated. Notice the occurrence of yellow squares in proximity with red and orange, which depicts delayed update. (b) shows the DAG produced as a result of the hierarchical update where edges, shown with arrows, signify the direction in which the influence was transmitted.

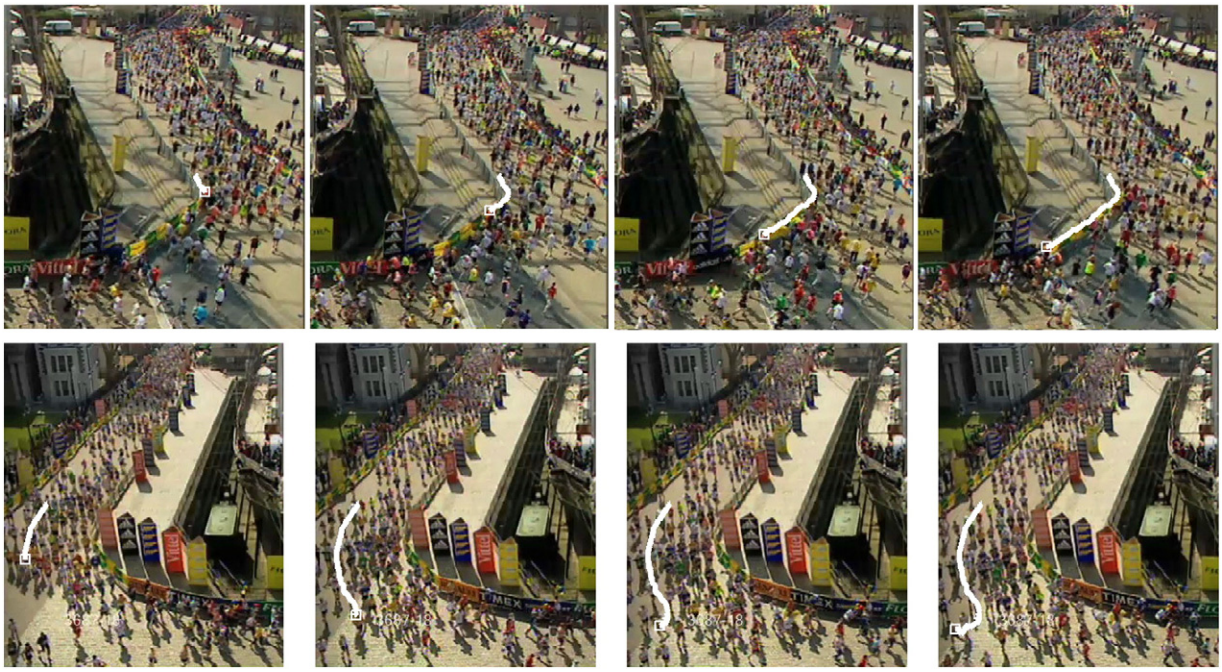

Fig. 9. Results of delayed update: The first row shows an anomaly where a person moves against the crowd flow. The second row shows results of tracking a particular individual who initially moved with the crowd but later decided to leave the marathon. This shows that instantaneous flow provides reasonable predictions when tracking anomalies. Note that, we do not detect anomalies per se, but whenever the appearance-based confidence from the underlying tracker is below τ, which may happen in the case of anomalies, we rely on instantaneous flow to provide predictions.

Although the proposed approach is complementary and an alternative to methods that track dense crowds after modeling crowd flow, for the sake of comparison, we report results using methods by Ali and Shah [16], who model crowd flow using various floor fields (FF), as well as Rodriguez et al. [34] who use Correlated Topic Model (CTM) to capture crowd behavior. The idea is to ensure that the performance without learning crowd flow remains comparable to the alternative approaches where crowd flow is modeled in advance, i.e., where data from the future is used to learn the behavior of the crowd. In addition, we compared against Park et al. [21] who use contextual information for tracking by solving a MRF framework using mean-shift belief propagation (MSBP). We also generated results from the template-based trackers such as mean-shift (MS) and Normalized Cross Correlation (NCC). NCC was used as the underlying tracker for the proposed method, as given by Eq. (6).

Both the previous methods [16,34] track individuals after manual initialization. The primary reason is to discard the effect of human detection which is extremely difficult for these sequences. We also manually initialized individuals by placing a fixed-sized square template around the initialization location. Template size for each sequence is given in the third row of Table 1. In addition, new individuals were initialized as they entered the scene. The queens were selected only when new initializations took place, which typically occurred after every fifty frames. The number of clusters for queen selection was set at $k = 100$ for all sequences. The templates were updated after every 10 frames and the value of $\tau = 0.90$ was selected. Therefore, if the value from the underlying tracker at peak location was greater than 0.90, the position of an individual was updated. A higher value for this threshold reduced the performance due to increased dependence on instantaneous flow, which is sometimes very noisy.

Fig. 10 shows the results obtained for the eight sequences. The first and third columns show the initial frame of each sequence with all the tracks output by the proposed method. In the second and fourth columns, we show graphs that reflect tracking accuracies of various methods. In these graphs, the x-axis shows the distance in pixels ranging from 0 to 25 and the y-axis is the percentage of tracked point from all trajectories that lie within that distance from the corresponding ground truth points. The curve from the proposed method is shown in red. The other methods are MSBP [21] shown in green, FF [16] in yellow, CTM [34] in orange, as well as baseline MS (mean-shift) in cyan and NCC (normalized cross-correlation) in blue. The values of these curves at 15 pixel threshold are given in Table 1. The proposed method performs equal or better than the comparison methods for Sequences 1–6. This illustrates that even without learning crowd flow, the prominence and spatial context are helpful enough to give decent tracking results. However, for Sequences 7 and 8, the results are lower by 1 to 2%, respectively. For Sequence 7, the reason is primarily the camera angle and large perspective distortion compared to other sequences. For Sequence 8, the reason lies in the density of the crowd. At lower densities, the individuals have more freedom to move, and thus, the motion of neighboring and prominent individuals is not a reliable estimate of the motion of a particular individual under consideration. Furthermore, the evaluation on these two sequences is also done on fewer people than the rest of the sequences.

Next, we present some qualitative results on Sequences 4 and 5. In Fig. 11, we show tracks of four different individuals from Sequence 5. The ground truth is shown in green, while track from the proposed method is in yellow. In Fig. 11(a–c), the track from the proposed method perfectly aligns with the ground truth, while Fig. 11(d) shows a failure case, where the track was lost soon after the initialization. The reason for this failure was that the person under consideration was wearing a dark-colored shirt. After a few frames of successful tracking, the person came into a position where he or she was surrounded by shadows of several other individuals. The underlying tracker confused the shadow with the person and started chasing the shadow. This highlights the importance of appearance in tracking dense crowds, since it can sometimes dominate auxiliary information provided in the form of better predictions.

Fig. 12 shows eight examples of tracks obtained from Sequence 4 using the proposed method, FF [16] and CTM [34]. In this figure, the

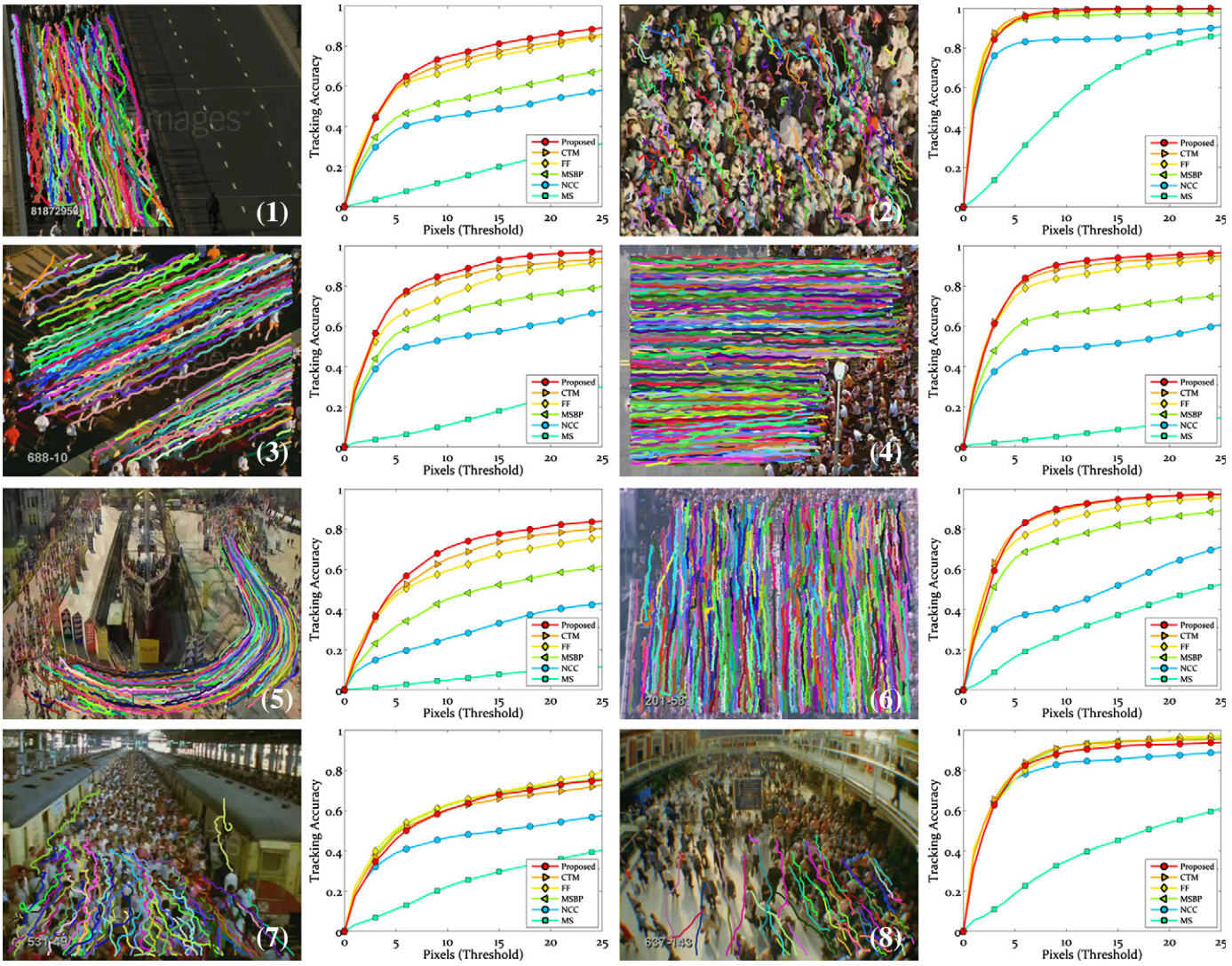

Fig. 10. Results on eight sequences used in our experiments: Tracks obtained from the proposed method are shown on the first frame of each sequence, shown in first and third columns. Graphs in the second and fourth columns show the tracking accuracies of baseline NCC (blue), MS (cyan), MSBP [21] (green), FF [16] (yellow), CTM [34] (orange), as well as the proposed method (red).

green track is manually-labeled ground truth, while yellow, orange and red tracks correspond to [16], [34] and the proposed method, respectively. An analysis of the erroneous tracks reveals that most of the id-switches were between people wearing the same color. The proposed method captures the constraints from neighbors which prohibit the jumping of the tracker across different people. The first row (Fig. 12(a,b)) shows instances where FF failed to track the individuals, whereas both CTM and the proposed method successfully tracked the individuals. The second row (Fig. 12(c,d)) shows instances where CTM failed, but FF and the proposed method were successful. The third row (Fig. 12(e,f)) shows instances where only the proposed method was successfully able to track the individuals. The last row shows an instance where all methods succeeded (Fig. 12(g)), and where all failed (Fig. 12(h)).

In order to test the contributions of various aspects of the proposed method, we ran a small experiment whose results are presented in Fig. 13. This plot shows that without the guidance of the queens and neighbors, i.e., using only self-component p_S in Eq. (1), the results are close to 70%; influence from neighbors, in the form of NMC with randomly initialized queens, adds 20% to tracking accuracy; while salient queens identified using Algorithm 1 add another 6%, giving 96% tracking accuracy of the proposed algorithm at the 10 pixel threshold. For this particular sequence, both prominence and NMC contribute to increase in tracking accuracy, however, this may not always be the case. For instance, prominence is of little value when all people have the same

Table 1
Quantitative comparison: Some statistics for the eight sequences are given in first three rows, while the last six rows are the results for the six methods. These are the values of curves in Fig. 10 at $T = 15$ pixels, which signifies the percentage of points in all tracks that lie within 15 pixels of ground truth. The numbers in bold indicate the best performance for each sequence among all the methods.

	Seq 1	Seq 2	Seq 3	Seq 4	Seq 5	Seq 6	Seq 7	Seq 8
# Frames	840	134	144	492	464	333	494	126
# People	152	235	175	747	171	600	73	58
Template size	14	16	14	16	8	10	10	10
NCC	49%	85%	58%	52%	33%	52%	50%	86%
MS	19%	67%	16%	8%	7%	36%	28%	43%
MSBP	57%	97%	71%	69%	51%	81%	**68%**	**94%**
FF	74%	99%	83%	88%	66%	90%	**68%**	93%
CTM	76%	**100%**	88%	92%	72%	**94%**	65%	**94%**
Proposed	**80%**	**100%**	**92%**	**94%**	**77%**	**94%**	67%	92%

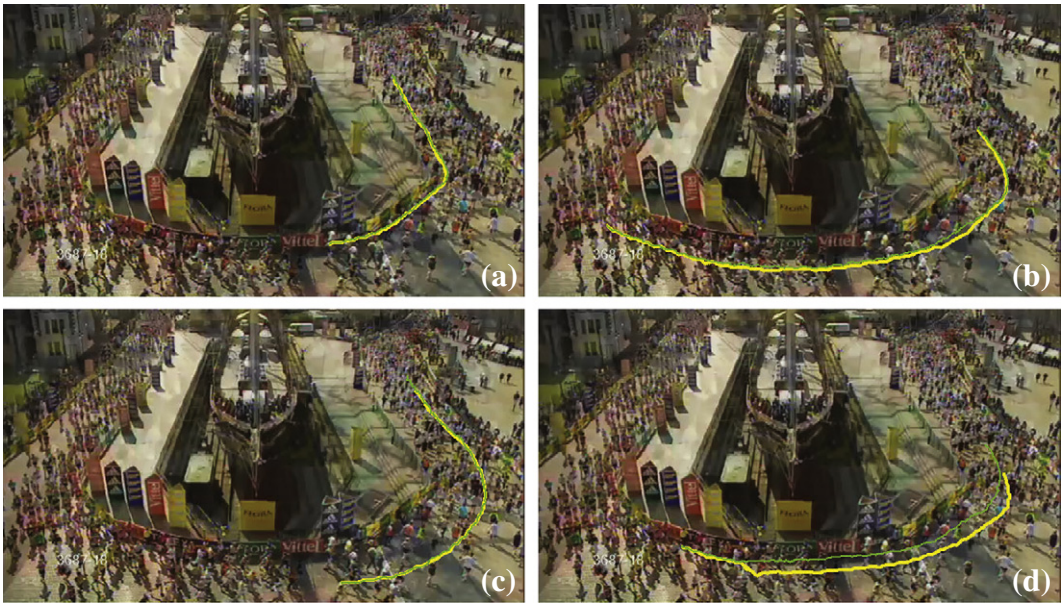

Fig. 11. Qualitative results on Sequence 5: This figure shows tracks of four different individuals from Sequence 5. The ground truth is shown in green, while track from proposed method is in yellow. In (a–c), the track from proposed method perfectly aligns with the ground truth, while (d) shows a failure case, where the track was lost soon after the initialization.

appearance or when everybody in the scene looks different and distinguishable. Similarly, the assumption of motion concurrence breaks at low densities when people have more freedom to move. However, it can be concluded from Figs. 10 and 13 that, while the contributions will vary for different scenes, in general, all components are necessary for an increase in tracking accuracy in structured dense crowds.

5. Conclusion

We introduced a novel method for tracking in dense crowds without using any prior knowledge about the scene, in contrast to previous works which always use some training and modeling of crowd flow using data from the past as well as the future. Beginning with prominent individuals, we track all individuals in the crowd in an ordered fashion employing influence from the neighbors and confidence from template-based tracker. We showed the performance of added functionality via scene-derived visual and contextual information, which significantly improved the template-based tracker. Future work will include making the proposed method more robust by using information from multiple frames at the same time.

Acknowledgment

This material is based upon work supported in part by the U. S. Army Research Laboratory and the U. S. Army Research Office under contract/grant number W911NF-09-1-0255.

Appendix A

Theorem. The number of times a target is visited (attempted for update) at time t is finite given $\tau < 1$.

Proof. The target *revisited* can either be a queen or a non-queen. After the first visit, in case the queen fails condition in Algorithm 2, Line 15, the algorithm, at later visits, will treat it as non-queen with prior probability governed by Eq. (5).

Let $p\left(z_i^{t,k}|\hat{x}_i^{t,k}\right)$, denote the probability of observation on *kth* visit to a non-queen target i at time t. There can be two cases at *kth* visit:

Case 1 $\forall\, j|j \in NN_i, \lambda_j = 1$.

Case 2 $\exists\, j|j \in NN_i, \lambda_j = 0$.

For Case 1, the distribution from Eq. (5) will not change for $l > k$. For Case 2, there are two possibilities: either for some $k' > k$, all the neighbors of the target get updated, which will collapse Case 2 to Case 1. The other possibility is when at least one of the neighbors is in the same situation as target i (i.e., $\lambda = 0$ for both). Under such circumstances, Eq. (5) for $l > k$ visits will still not change, since λ_j will be zero in Eq. (2), thus, the influence from neighbor j will not be used for updating state of target i.

In either case, $\exists k'|\forall k l > k'$,

$$p\left(x_i^{t,l}|x_{j\in NN_i}^{t}, x_i^{t-1}\right) = p\left(x_i^{t,k'}|x_{j\in NN_i}^{t}, x_i^{t-1}\right). \tag{9}$$

It follows from Algorithm 2, Line 15 that for

$$0 < \eta_i \leq \left\lceil \frac{\log\left(p(z_i^{t,k'} \mid \hat{x}_i^{t,k'})\right)}{\log \tau} \right\rceil - 1 < \infty \tag{10}$$

state of target i will be updated by Algorithm 2.

References

[1] B. Zhan, D. Monekosso, P. Remagnino, S. Velastin, L.-Q. Xu, Crowd analysis: a survey, Journal of Machine Vision and Applications 19 (2008) 345–357.
[2] V. Rabaud, S. Belongie, Counting crowded moving objects, IEEE Conference on Computer Vision and Pattern Recognition, 2006.
[3] X. Wu, G. Liang, K.K. Lee, Y. Xu, Crowd density estimation using texture analysis and learning, IEEE International Conference on Robotics and Biomimetics, 2006.
[4] W. Ge, R. Collins, B. Ruback, Automatically detecting the small group structure of a crowd, IEEE Workshop on the Applications of Computer Vision, 2009.
[5] R. Mehran, A. Oyama, M. Shah, Abnormal crowd behavior detection using social force model, IEEE Conference on Computer Vision and Pattern Recognition, 2009.
[6] L. Kratz, K. Nishino, Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models, IEEE Conference on Computer Vision and Pattern Recognition, 2009.

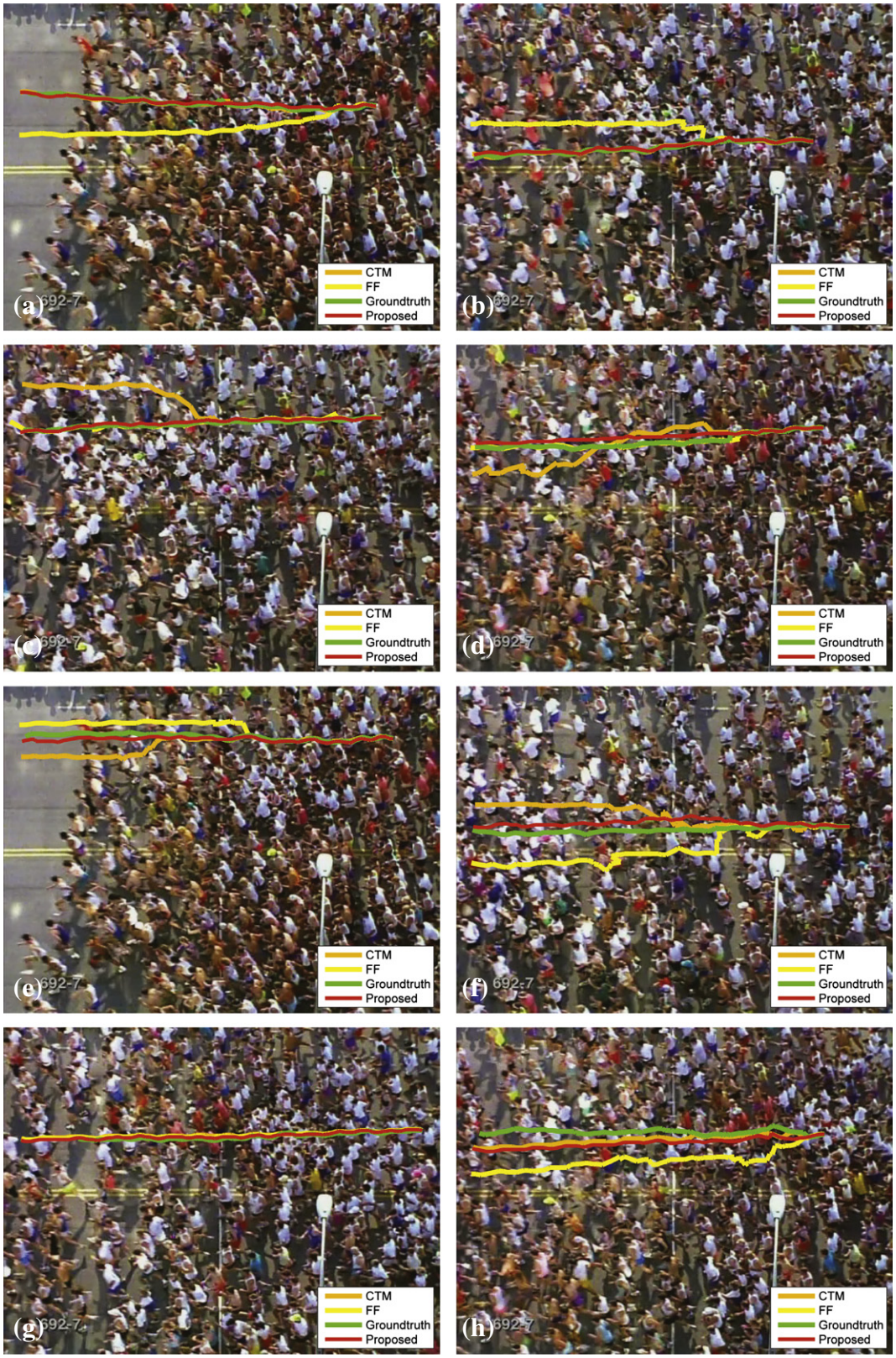

Fig. 12. Eight examples from Sequence 4 that show the comparison of the proposed method (red) with FF [16] (yellow) and CTM [34] (orange). The ground truth track is depicted in green. (a,b) show instances where FF failed but CTM and proposed method succeeded. (c,d) show instances where CTM failed but the other two succeeded. (e,f) show instances where both FF and CTM failed but the proposed method succeeded. Finally, (g) shows a common instance where all trackers successfully tracked the individual, while (h) shows a rare case where all three failed.

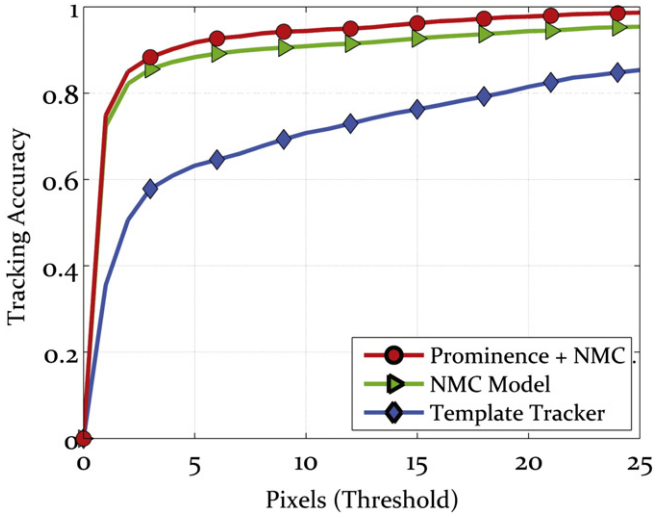

Fig. 13. Contribution towards tracking accuracy by major components of the algorithm: The experiment was done on sequence 2 at 2.5 fps. The *x*-axis is the distance threshold in pixels, while the *y*-axis is the percentage of tracked points that lie within that distance from the ground truth. This shows that all aspects are important for improvement in tracking results.

[7] T. Cao, X. Wu, J. Guo, S. Yu, Y. Xu, Abnormal crowd motion analysis, IEEE International Conference on Robotics and Biomimetics, 2009.
[8] V.B.V. Mahadevan, W. Li, N. Vasconcelos, Anomaly detection in crowded scenes, IEEE Conference on Computer Vision and Pattern Recognition, 2010.
[9] S. Ali, M. Shah, A Lagrangian particle dynamics approach for crowd flow segmentation and stability analysis, IEEE Conference on Computer Vision and Pattern Recognition, 2007.
[10] D. Lin, E. Grimson, J. Fisher, Modeling and estimating persistent motion with geometric flows, IEEE Conference on Computer Vision and Pattern Recognition, 2010.
[11] D. Sugimura, K. Kitani, T. Okabe, Y. Sato, A. Sugimoto, Using individuality to track individuals: clustering individual trajectories in crowds using local appearance and frequency trait, IEEE International Conference on Computer Vision, 2009.
[12] Z. Wu, N. Hristov, T. Hedrick, T. Kunz, M. Betke, Tracking a large number of objects from multiple views, IEEE International Conference on Computer Vision, 2009.
[13] X. Song, X. Shao, H. Zhao, J. Cui, R. Shibasaki, H. Zha, An online approach: learning-semantic-scene-by-tracking and tracking-by-learning-semantic-scene, IEEE Conference on Computer Vision and Pattern Recognition, 2010.
[14] M. Hu, S. Ali, M. Shah, Learning motion patterns in crowded scenes using motion flow field, International Conference on Image Processing, 2006.
[15] A. Chan, N. Vasconcelos, Counting people with low-level features and Bayesian regression, IEEE Trans. Image Process. 21 (2012) 2160–2177.
[16] S. Ali, M. Shah, Floor fields for tracking in high density crowd scenes, European Conference on Computer Vision, 2008.
[17] D. Helbing, P. Molnár, Social force model for pedestrian dynamics, Phys. Rev. E 51 (5) (1995) 4282–4286.
[18] M. Yang, Y. Wu, S. Lao, Intelligent collaborative tracking by mining auxiliary objects, IEEE Conference on Computer Vision and Pattern Recognition, 2006.
[19] M. Yang, Y. Wu, G. Hua, Context-aware visual tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009.
[20] Z. Khan, T. Balch, F. Dellaert, An MCMC-based particle filter for tracking multiple interacting targets, European Conference on Computer Vision, 2004.
[21] M. Park, Y. Liu, R.T. Collins, Efficient mean shift belief propagation for vision tracking, IEEE Conference on Computer Vision and Pattern Recognition, 2008.
[22] R.T. Collins, Y. Liu, On-line selection of discriminative tracking features, IEEE International Conference on Computer Vision, 2003.
[23] V. Mahadevan, N. Vasconcelos, Saliency-based discriminant tracking, IEEE Conference on Computer Vision and Pattern Recognition, 2009.
[24] R.A. Smith, Density, velocity and flow relationships for closely packed crowds, Saf. Sci. 18 (4) (1995) 321–327.
[25] Z. Fang, S.M. Lo, J.A. Lu, On the relationship between crowd density and movement velocity, Fire Saf. J. 38 (3) (2003) 271–283.
[26] M. Moussaid, N. Perozo, S. Garnier, D. Helbing, G. Theraulaz, The walking behaviour of pedestrian social groups and its impact on crowd dynamics, PLoS ONE 5 (2010) e10047.
[27] N. Pelechano, J.M. Allbeck, N.I. Badler, Controlling individual agents in high-density crowd simulation, ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2007.
[28] S. Pellegrini, A. Ess, K. Schindler, L. van Gool, You'll never walk alone: modeling social behavior for multi-target tracking, IEEE International Conference on Computer Vision, 2009.
[29] K. Yamaguchi, A.C. Berg, L. Ortiz, T.L. Berg, Who are you with and where are you going? IEEE Conference on Computer Vision and Pattern Recognition, 2011.
[30] R. Garg, D. Ramanan, S.M. Seitz, N. Snavely, Where's Waldo: matching people in images of crowds, IEEE Conference on Computer Vision and Pattern Recognition, 2011.
[31] L. Kratz, K. Nishino, Tracking with local spatio-temporal motion patterns in extremely crowded scenes, IEEE Conference on Computer Vision and Pattern Recognition, 2010.
[32] L. Kratz, K. Nishino, Going with the flow: pedestrian efficiency in crowded scenes, European Conference on Computer Vision, 2012.
[33] L. Kratz, K. Nishino, Tracking pedestrians using local spatio-temporal motion patterns in extremely crowded scenes, IEEE Trans. Pattern Anal. Mach. Intell. 34 (2012) 987–1002.
[34] M. Rodriguez, S. Ali, T. Kanade, Tracking in unstructured crowded scenes, IEEE Conference on Computer Vision and Pattern Recognition, 2009.
[35] M. Rodriguez, J. Sivic, I. Laptev, J.-Y. Audibert, Data-driven crowd analysis in videos, IEEE International Conference on Computer Vision, 2011.
[36] L. Leal-Taixé, G. Pons-Moll, B. Rosenhahn, Everybody needs somebody: modeling social and grouping behavior on a linear programming multiple people tracker, 1st ICCV Workshop on Modeling, Simulation and Visual Analysis of Large Crowds, 2011.