Floor Fields for Tracking in High Density Crowd Scenes

Saad Ali, Mubarak Shah

Computer Vision Lab, University of Central Florida, Orlando, USA
sali,shah@eecs.ucf.edu

Abstract. This paper presents an algorithm for tracking individual targets in high density crowd scenes containing hundreds of people. Tracking in such a scene is extremely challenging due to the small number of pixels on the target, appearance ambiguity resulting from the dense packing, and severe inter-object occlusions. The novel tracking algorithm, which is outlined in this paper, will overcome these challenges using a *scene structure based force model*. In this force model an individual, when moving in a particular scene, is subjected to global and local forces that are functions of the layout of that scene and the locomotive behavior of other individuals in the scene. The key ingredients of the force model are three floor fields, which are inspired by the research in the field of evacuation dynamics, namely *Static Floor Field* (SFF), *Dynamic Floor Field* (DFF), and *Boundary Floor Field* (BFF). These fields determine the probability of move from one location to another by converting the long-range forces into local ones. The SFF specifies regions of the scene which are attractive in nature (e.g. an exit location). The DFF specifies the immediate behavior of the crowd in the vicinity of the individual being tracked. The BFF specifies influences exhibited by the barriers in the scene (e.g. walls, no-go areas). By combining cues from all three fields with the available appearance information, we track individual targets in high density crowds.

1 Introduction

Tracking individuals in a high density crowd scene is challenging for a number of reasons: 1) the number of pixels on an object decreases with the increasing density of the objects; 2) constant interaction among the individuals in a crowd makes it hard to discern individuals from one another; 3) occlusions caused by inter-object interactions result in the loss of observation of the target object; 4) the mechanics of a human crowd is complex as it exhibits goal-directed dynamics and psychological characteristics which in turn influence how an individual person will behave in a crowd. Examples of high density crowd scenes are shown in Fig. 1.

To overcome some of these challenges, we have developed an algorithm for tracking individual targets in *high density crowd scenes* containing hundreds of people at a time. The proposed algorithm is based on the observation that *the locomotive behavior of an individual in a crowded scene is a function of*

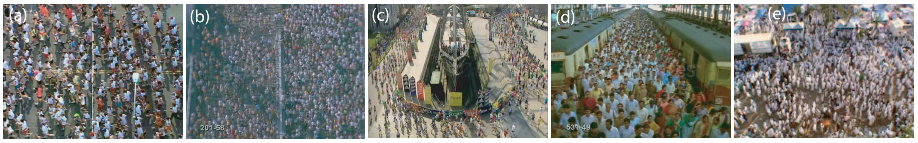


Fig. 1. Examples of high density crowd scenes. (a)-(c) Hundreds of people participating in marathons. (d) A scene from a densely packed railway station in India. (e) A group of people moving in opposite directions.

collective patterns evolving from the space-time interactions of individuals among themselves and with the layout of the scene. These collective behavioral patterns, therefore, can be included as an auxiliary source of information to constrain the likely locations or paths that can be taken by the target object in the scene. In other words, natural crowd flow and scene constraints influencing the behavior of a person in a dense crowd can be used as priors to impose high-level direction for tracking purposes. Our novel model called the 'scene structure force model', is proposed to directly incorporate such prior knowledge and influences.

In our tracking algorithm, the crowd is treated as a collection of mutually interacting particles. This is a reasonable model, because when people are densely packed individual movement is restricted and members of the crowd can be considered granular particles. In order to track a specific individual in the crowd, we model the instantaneous movement of that person (or particle) with a matrix of preferences containing the probabilities of a move in a certain preferred direction. The size of the matrix is a function of the maximum allowed velocity, i.e. it is determined by the number of cells the target individual is allowed to move into any direction with the center element of the matrix corresponding to no motion. The probabilities take into consideration multiple sources of information and constraints arising from the appearance of the target individual, flow of the crowd and the structure of the scene. The flow of the crowd and scene structure incorporated by introducing the concept of *floor fields*, which model the interactions between individuals and their preferred direction of movement by transforming long ranged forces into local ones. For instance, a long range force that compels an individual in a crowd to move towards the exit door, can be converted into a local force such that it increases the instantaneous probability of a move in that direction. The transition probability of a tracked person then depends on the strength of the floor field in his/her neighborhood. The concept of a floor field is inspired by the field of evacuation dynamics [1, 2], where floor fields are *manually* designed to simulate behaviors of pedestrians in panic situations. We *automatically* compute three floor fields from the visual data: a 'Static Floor Field' (SFF), a 'Boundary Floor Field' (BFF), and a 'Dynamic Floor Field' (DFF).

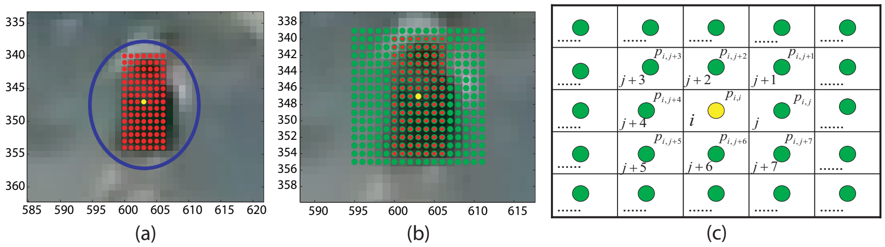
2 Related Work

Tracking is a popular research area in computer vision [16, 6]. In this section, we briefly review tracking algorithms that were designed specifically for crowded scenes. Interested readers are referred to the recent survey by Yilmaz *et al.* [10] for an in-depth review of the tracking literature. For tracking in crowded environments, the method proposed by Zhao *et al.* [14] was one of the first algorithms. Their algorithm used articulated ellipsoids to model the human shape, color histograms to model appearance, and a Gaussian distribution to model the background for segmentation. The initial detection of objects was based on the their segmentation scheme ([13]) proposed earlier. However, their method is not suitable for the high density crowd situations that we are dealing with, because in such a situation the whole human body is rarely visible, which makes it impossible to fit elaborate ellipsoid models of human body shapes. Brostow *et al.* [12] presented a probabilistic framework for the clustering of feature point trajectories for detection and tracking of individual pedestrians in crowds. They hypothesized that pairs of points that appear to move together were likely to be part of the same individual and as such could be used for detection and tracking.

There are other interesting tracking methods which were developed for tracking sparse crowds of ants [6], hockey players [9], crowds of clumped people [8], a dense flock of bats [4], and biological cells [5]. In general, these tracking methods are object centric and do not exploit any high level or global knowledge that may aid the tracking algorithm. This is one of the major difference between our tracking algorithm and these approaches; we have integrated high level constraints resulting from crowd flow and scene structure into the tracking algorithm.

3 Tracking Framework

The crowd flow in the scene is treated as a collection of mutually interacting particles. Therefore, given a video $E = [f_1, f_2, \ldots, f_N]$, where N is the total number of frames, the image space is discretized into cells where each cell is occupied by a single particle $o_{\mathbf{x}i}$. Here, $\mathbf{x}i = (xi, yi)$ is the coordinate of the i-th pixel at which the particle is located. The relationship between cells, pixels and particles is as follows: Cells form a grid over the spatial extent of the image and each cell is always associated with a single particle, although a cell can contain more than one pixel depending upon the resolution of the grid. A particle represents all the pixels in the cell. For tracking, the target individual is represented by a set of particles $\mathcal{P} = [\ldots, o_{\mathbf{x}i}, \ldots]$ (red particles in Fig. 2(a)) as an individual can span multiple cells in the image. An appearance template, H, of the target is then computed using all the pixels corresponding to the underlying pixels represented by particles $o \in \mathcal{P}$. The target moves from one cell to the next at discrete time steps, $t \rightarrow t + 1$, according to a transition probability that determines the likely direction of the motion. This transition probability is associated with the centroid (yellow particle in Fig. 2(a)) although its computation uses information from all the particles in the set $\mathcal{P}$. The transition probability is determined by two factors: 1) the similarity between the appearance templates at the current location and the next; 2) the influence generated by the floor fields. Formally, if the target individual is currently at cell i (the cell containing the yellow particle in Fig. 2(c)), then the probability of moving to a neighboring cell j (cells



(a) (b) (c)

Fig. 2. (a) The particles $o \in \mathcal{P}$ belonging to the individual we want to track. Each particle occupies a single cell with the yellow particle representing the centroid of the object. In order to track the target, we track the yellow particle through the scene. (b) The green particles represent the search area for the possible next location of the yellow particle. (c) The matrix of preferred walking directions around the yellow particle. Each value in the matrix is the probability of the yellow particle moving from the center cell i to the surrounding cell. The transition probability p_{ij} is computed using 1.

containing green particles in Fig. 2(c)) is:

$$p_{ij} = Ce^{k_D D_{ij}} e^{k_S S_{ij}} e^{k_B B_{ij}} R_{ij}, \tag{1}$$

where D_{ij}, S_{ij}, and B_{ij} are the influences of the DFF, SFF, and BFF, respectively. While k_D, k_S, and k_B are the coupling strength of the tracked object to the DFF, SFF, and BFF, respectively. R_{ij} is the similarity measure between the initial appearance template H and the new appearance template of the target computed at location j. C is a normalization constant.

The benefit of using the above equation is that it employs the crowd flow and scene layout constraints to weight the appearance similarity information when tracking the person. The crowd flow constraints are captured by the DFF and SFF, while the scene layout constraints are captured by the BFF. The weighting can be understood in the following manner: if the crowd follows a path say from the cell i to j, the SFF will have a high value for S_{ij} and in turn will favor that direction of motion. This will increase the likelihood of matching the current appearance template of the target at location i with the one at location j at next time instance. Similarly, if immediate crowd behavior is such that it is moving from the cell i to $j+1$, the DFF will have a higher value for $D_{i,j+1}$ and will favor that direction of motion. This will increase the likelihood of matching the appearance template at location i with the one at location $j+1$. Thus we can use the crowd flow information to gain more confidence about our appearance matching scores. Next, we describe the algorithm for computing S_{ij}, D_{ij}, and B_{ij} from their respective floor fields.

3.1 Static Floor Field - S_{ij}

The SFF is aimed at capturing attractive and constant properties of the scene. These properties include preferred areas, such as dominant pathes often taken by the crowd as it moves through the scene, and preferred exit locations. In our framework, the SFF is computed only once for a given scene during the

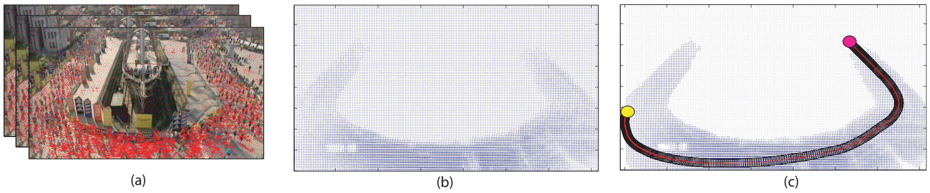


Fig. 3. (a) The dense optical flow for frames, $[f_1, f_2, \ldots, f_M]$, of the video. (b) The computed point flow field. (c) The sink seeking process: the yellow circle represents the initial location, while the red circle shows the corresponding sink. The black windows represent the area used to weight the local velocity and propagate the sink seeking process. The red trajectory represents the 'sink seeking path', while the number of black windows represents the corresponding number of sink steps.

learning period, which spans initial $M << N$ frames. The steps involved in the computation of SFF are as follows: i) Computation of a point flow field; ii) Sink Seeking.

Point Flow Field: A 'point flow field' represents the instantaneous changes of motions present in the video. Each vector in this field is a 4-dimensional vector obtained by augmenting the local flow vector with the position information. The new vector is referred to as a 'point flow vector', hence the name 'point flow field'. Using the first M frames of an input video, $E = [f_1, f_2, \ldots, f_M]$, a dense optical flow can be computed between consecutive frames using the method of [15]. Then, for each cell (or pixel) i, a point flow vector, $Z_i = (X_i, V_i)$, is computed, which includes both the location $X_i = (x_i, y_i)$ and the optical flow vector $V_i = (v_{x_i}, v_{y_i})$. Note that V_i is the mean of $(M - 1)$ optical flow vectors computed at pixel i from the first M frames of the video. All flow vectors averaged over M frames of the video then constitute the 'Point Flow Field', which represents the smoothed out motion information of the video in that interval. This smoothed motion information assists in computing the dominant properties (paths, exits) of the scene, which is the primary objective of the SFF. Fig. 3(a) shows flow vectors generated for a marathon video using the dense optical flow computation [15]. The resulting point flow field is given in Fig. 3(b).

Sink Seeking Process: The point-flow field is then used to discover the regions in the scene called 'sinks'. The idea behind the sink seeking process is that the behavior of large crowds of pedestrians in locations such as sporting events, religious festivals, and train-stations can be described as goal directed and rational. That is the members of the crowd have clear knowledge of what and where their goals lie in the scene [3]. Therefore, if we know the locations of the sinks, which are the desired goals (or locations), then for any given point in the scene we can compute a local force representing the tendency of the individual at that point to move towards the nearest sink. This local force will be a function of the shortest distance to the sink in terms of the appropriate distance metric.

In order to compute the sinks and their shortest distances, we initialize a grid of particles over the point flow field of the scene. Then, a particle dropped at a non-zero velocity location has the tendency to move to a new position under the influence of the neighboring point flow vectors. It then moves from that

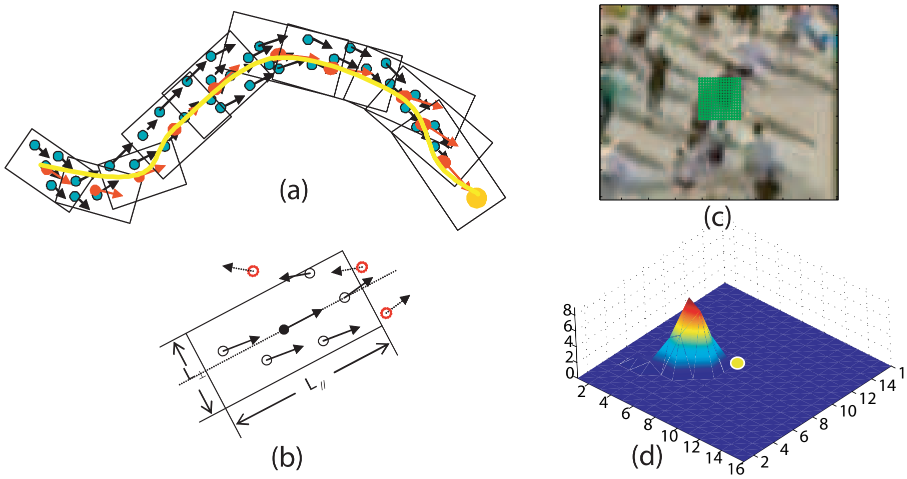


Fig. 4. (a) Sink seeking (red: the states of the point flow in the sink seeking process, orange: the sink, rectangles: sliding windows, yellow: the sink path); (b) Sliding window (solid circle: the point flow under consideration; rectangle: sliding window; hollow circles: neighboring points; red circles: non-neighboring points).(c) The region where we want to compute the DFF. (d) The computed DFF where the yellow circle represents the cell i. In this case, the DFF is capturing the dynamic relationship between the cell i and the neighboring cells.

new position to the next one and continues this process. in order to optimally combine the influence of the neighboring point flow vectors, the velocity at each new position is re-estimated as the weighted sum of its neighboring velocities (Fig. 4(a)). The weights are computed using a kernel density method. If all the weights are below a threshold, it implies that the new velocity is not significant enough to drive the particle to the next position. Therefore, the particle will stop and the process of pursuing a new location is discontinued. We call this process the *sink seeking process* and the last state (stopping state) of the process is called the *sink*. The corresponding path taken by the particle to reach the sink is called the *sink path*(Figures 3(c) and 4(a)). The length of the sink path is the minimum number of steps required to reach the closest exit location in the scene. The number of steps taken during the sink-seeking process to reach the sink is called the *seek steps*. This is also the distance metric used for representing the shortest distance. Note that the sink seeking process is carried out for each point in the point-flow field, thus generating one sink path per point. Formally, the 'sink seeking process' can be described as follows: Suppose $\{Z_1, Z_2, \cdots, Z_n\}$ is the point flow field of the video, where the state of the point i is defined as: $\widetilde{Z}_{i,t} = (\widetilde{X}_{i,t}, \widetilde{V}_{i,t}), t = 1, 2, ...,$ and computed as:

$$\widetilde{Z}_{i,1} = Z_i \ , \quad \widetilde{X}_{i,t+1} = \widetilde{X}_{i,t} + \widetilde{V}_{i,t}, \tag{2}$$

$$\widetilde{V}_{i,t} = \frac{\sum_{n \in Neighbor(\widetilde{X}_{i,t})} V_n W_{t,n}}{\sum_{n \in Neighbor(\widetilde{X}_{i,t})} W_{t,n}} \ , \ W_{t,n} = \exp\left(-\left\| \frac{\widetilde{V}_{t-1} - V_n}{h_{t-1}} \right\|^2\right), \tag{3}$$

In the previous equations, it is clear that the new position of a point only depends on the location and velocity at the previous state. However the new velocity

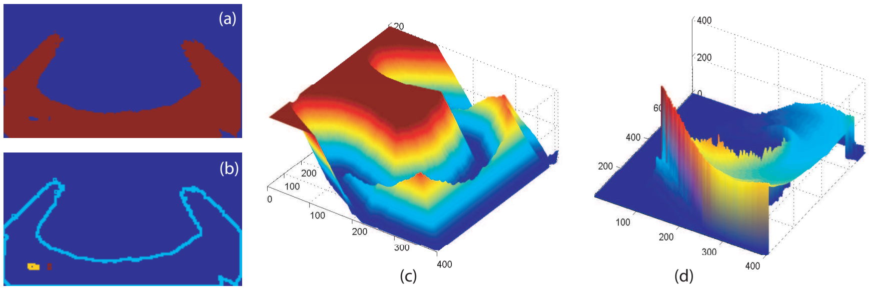


Fig. 5. (a) Crowd flow segmentation obtained by the method described in [11]. (b) The edge map obtained from the segmentation. (c) The boundary floor field for the sequence shown in Fig. 1c. The higher values in the field represent the decreasing effect of the repulsive potential generated by the barriers. In this case, the barrier effect vanishes for distances greater than 20 pixels. (d) The static floor field computed by our algorithm for the sequence shown in Fig. 1(c).

$\widetilde{V}_{i,t+1}$ depends not only on the previous velocity, but also on the observed velocities of its neighbors, which represent the motion trend of a local group. In this work, we employ the kernel based estimation which is similar to the mean shift approach [16]. However, there is one important difference. In mean shift tracking, the *appearance* of pixels within a small neighborhood of the object is used to determine the location of the object in the next frame. In our approach, we use *the location and the velocity* of the neighboring points in the point flow field to determine the next location. There are other methods proposed in the literature for locating sources and sinks in the scene ([7]), however they do not provide the shortest distance to sink for each location in the scene. This distance is essential for our algorithm to compute the local SFF force. Another point that we like to highlight is that previous approaches only use the sources/sinks to initiate person hypotheses for tracking without taking advantage of statistics of what tends to happen between sources and sinks. Through the sink seeking process we are able to capture the behavior between sources and sinks as well.

SFF Generation: The SFF is generated by using the sink steps for each sink path. We find the location (x, y) (in the image space) at which each sink path starts and place the value of corresponding 'sink step' at that location. Fig. 5(d) shows the computed SFF for the sequence in Fig. 1(c). It is interesting to note that the shape of the SFF emphasizes the notion that if you place a particle at any location, it will roll down towards the exit. This is precisely what the goal oriented dynamics of the crowd in this scene represent. In the tracking algorithm, the shape of the SFF translates into a force in the direction that requires the least number of steps to reach the nearest exit or sink. That is, the difference between the values in cell i and j in this field is the measure of the S_{ij} parameter of 1. Other SFFs are shown in Fig. 6.

3.2 Boundary Floor Field - B_{ij}

The purpose of the BFF is to capture influences generated by barriers or walls in the scene. This influence is usually repulsive in nature. The computation of the BFF requires the localization of physical and virtual barriers in the scene. The virtual barriers arise from the presence of dynamically distinct crowd flows in the scene. The computation of the BFF is performed after a set time interval ΔT_B, and works on a group of frames defined by the parameter N_B. That is, computation of BFF at time t uses frames $[f_t, f_{t+1}, \ldots, f_{t+N_B}]$.

The computation of BFF is based on the crowd flow segmentation algorithm proposed earlier by the authors [11]. In that algorithm, physical and virtual barriers in the scene were represented by the ridges of the Finite Time Lyapunov Exponent (FTLE) Field. The FTLE field is then used to compute a segmentation map in which different labels represented different crowd flow segments. In order to generate the BFF, we use this segmentation map and compute an edge map by retaining only the boundary pixels of each segment. Next, the shortest distance between the wall/barrier and each pixel is determined by computing the distance transform of this edge map (see Figs. 5a-c). Note that when a distance is larger than a certain threshold, the barrier effect vanishes completely. This vanishing effect is represented by the flattening of the surface (the red region) in Fig. 5(c). The difference between the values in cell j and i represent the measure of B_{ij} in 1. Examples of BFFs are presented in Fig. 6.

3.3 Dynamic Floor Field - D_{ij}

The objective of the DFF is to determine the behavior of the crowd around the individual being tracked. The instantaneous information about the motion of the crowd is an important cue for constraining likely future locations. In our framework, the instantaneous interaction among the members of the crowd is extracted by using a particle based representation. For a given scene, the DFF is computed at each time period by using a sliding window of N_D frames. That is, for computing the DFF at time t, we use frames $F_D = [f_t, f_{t+1}, \ldots, f_{t+N_D}]$. We

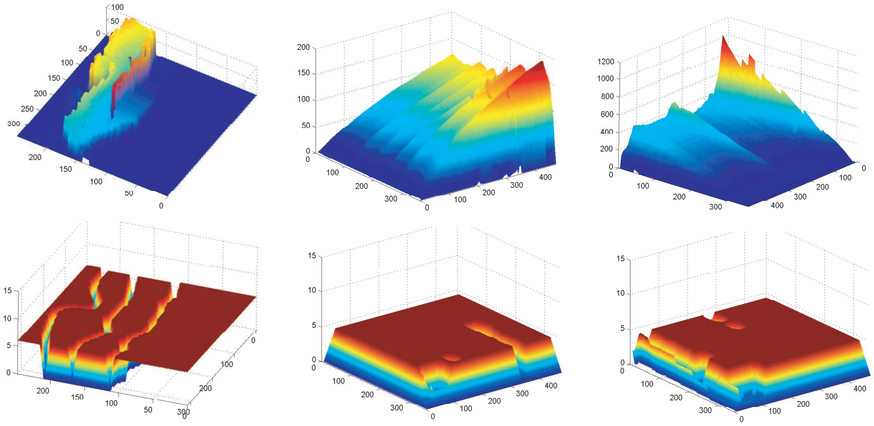


Fig. 6. The SFFs (top) and BFFs (bottom) of various sequences. Left: For the sequence in Fig. 1(e). Center: For the marathon sequence in Fig. 1(a). Right: For the marathon sequence in Fig. 1(b).

Fig. 7. Chips used for tracking. (a) Marathon-1. (b) Marathon-2. (c) Marathon-3.

first compute the optical flow between consecutive frames in F_D and stack them together to generate a 3D volume of optical flow fields. Next, a grid of particles is overlaid on the first flow field of the volume and numerically advected [11]. During the advection, whenever a particle jumps from a cell i to one of the neighboring cells j the value of interaction between these cells increases by one. The DFF can only have non-negative integer values and there is one DFF per cell where each DFF captures the dynamic interaction between the target cell i and remaining cells in the scene. A visualization of the DFF is shown in Fig. 4(c)-(d). Since the DFF is meant to capture the local interaction of particles around the tracked individual, Fig. 4(c)-(d) represents the shape of the DFF, but only in that local neighborhood. The peak in Fig. 4(d) represents the location where most particles end up if they pass through the cell containing the yellow particle.

4 Experiments and Discussion

A detailed experimental analysis was performed on the three marathon sequences shown in Figure 6. In addition, qualitative results are shown for a busy train-station sequence. In all the experiments, tracking began by selecting a rectangular region around the target object and using it to compute the gray-level appearance template. At each time instant, the next position of the target was chosen according to Equation 1, where the matrix of preferences around the current target location were twice the size of the selected rectangular region. We do not adapt the size of the window during the tracking. The appearance similarity was computed using normalized cross correlation and the template was progressively updated at each time instant. We set the values of k_S, k_D, and k_B equal to 0.02 for all experiments. The tracking results were stable for small changes in the values of these coupling factors. We used the first 50 frames of each sequence to construct the SFF. To compute the BFF and DFF, the values of $N_B = 20$ and $N_D = 5$ were used.

Marathon-1: This sequence (Fig. 1a) captures participants in a marathon from an overhead camera. It is a difficult sequence due to the severe occlusion among the participants, and the similar looking outfits worn by most of the athletes. The sequence has 492 frames, but each athlete remains in the field of view, on average, for 120 frames. We manually selected 199 individuals, shown in Fig. 7(a), from various frames for tracking. The average size of the selected chip was 14 × 22 pixels.

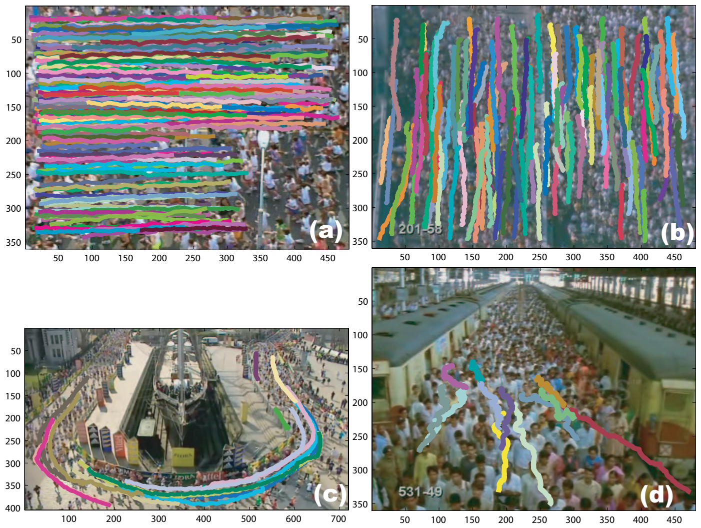


Fig. 8. Displays trajectories of individuals who were successfully tracked by our method. (a) Marathon-1. (b) Marathon-2. (c) Marathon-3. (d) Train Station.

A set of trajectories generated by our tracing algorithm is shown in Fig. 8(a). In total, we were able to successfully track 143 out of 199 individuals, i.e. the correct label was maintained throughout the time duration for which the athlete was visible in the FOV. Quantitative analysis of the tracking was performed by generating ground-truth trajectories for 50 athletes, which were selected randomly from the initial set of 199 athletes. The ground-truth is generated by manually tracking the centroid of each selected athlete. The ground-truth shows that these 50 athletes were visible for an average of 77 frames, and our algorithm tracked them for an average of 72 frames. This is summarized by the first 50 bars in the graph of Fig. 9. The average tracking error is summarized by the first 50 green bars in the graph of Fig. 12(a). The tracking error in a given frame is defined as the distance in pixels between the centroid of the object in the ground-truth and the centroid returned by our tracking algorithm. The tracking failure on this sequence (10(a)) occurred in situations where the target was completely occluded by either another athlete or the street-light in the scene. Since we did not use any prediction mechanism, we could not recover from full occlusion. However, partial occlusions were handled by our tracker.

Marathon-2: This sequence (Fig. 1(b)) also involves a marathon. However, the camera in this sequence is installed on a high-rise building to increase the FOV. As a result, the number of pixels on each individual is fewer. In addition, there are drastic illumination changes when athletes move into the shadow of the neighboring buildings. This sequence has 333 frames. We manually selected 120 individuals (Fig. 7(b)) from various frames for tracking. The average size of the selected chip was 13 × 16 pixels.

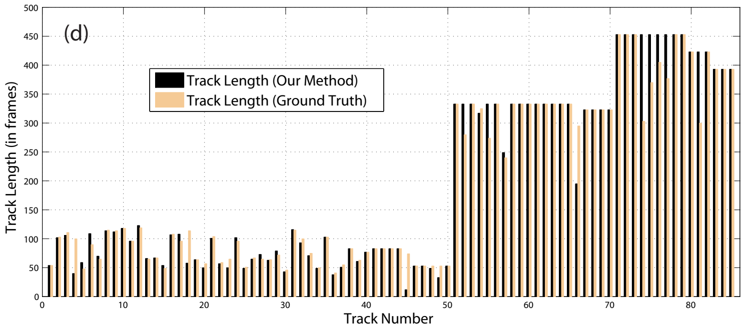


Fig. 9. A comparison of track lengths against the ground-truth: 1 to 50 Marathon-1; 51-70 Marathon-2; 71-85 Marathon-3.

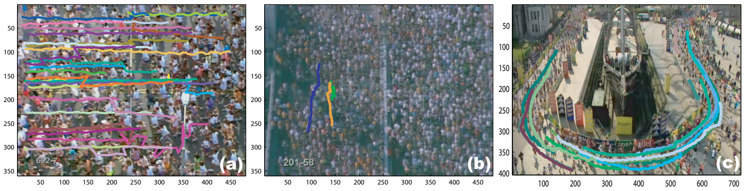


Fig. 10. The failure cases. (a) Marathon-1. (b) Marathon-2. (c) Marathon-3.

A set of trajectories generated by our tracing algorithm is shown in Fig. 8(b). In total, we were able to track 117 of the 120 (97.5%) individuals. A quantitative analysis was performed by generating ground truths for 20 of the athletes. The length of ground-truth trajectories and trajectories generated by our tracker is summarized by bars 51-70 in Fig. 9. The average tracking error is summarized by the green bars (51-70) in the graph of Fig. 12(b). It can be observed that our tracking was very accurate, in most cases, and able to overcome the illumination changes with the aid of the DFF and SFF. Some of the tracking failures on this sequence are shown in Fig. 10(b). These tracking failures were the result of severe illumination variation causing change in the appearance of the target.

Marathon-3: The third sequence (Fig. 1(c)) is extremely challenging due to two factors: 1) appearance drastically changes due to the U-shape of the path; 2) the number of pixels on target varies due to the perspective effect. The fewer pixels make it more difficult to resolve even partial occlusions. The sequence is 453 frames long. We manually selected 50 individuals (Fig. 7(c)) for tracking. The average size of the selected chip was 14 × 17 pixels. In total, we were able to track 38 of the 50 (76%) individuals without any tracking error (Fig. 8(c)). A comparison of track lengths returned by our algorithm with 15 ground-truth trajectories is given in Fig. 9. The average tracking error is summarized by bars 71-85 in Fig. 12(a). In addition, we performed tracking on a busy train-station sequence shown in Fig. 1(d). There, we tracked 20 individuals and some qualitative results are shown in Fig. 8(d).

Analysis: One of our experiments is now discussed in detail to provide an intuitive insight on how the floor fields help in improving tracking. For this purpose, we picked a track from Marathon-3, where the athlete was wearing a black shirt and running away from the camera (Fig. 2(a)). During the course of tracking, the athlete's appearance became ambiguous as neighboring athletes were also wearing black shirts. Fig. 11(a) (top-left) shows the surface of appearance similarity measure, which was obtained by matching the template in a 16×16 neighborhood, for one of those instances. The surface was relatively flat, showing a lack of a good match for the tracked person in the current frame. If we were to use only this surface, there was a high probability that the tracker would jump onto one of the neighboring athletes wearing the similar clothing. However, floor fields helped resolve this ambiguity, which is apparent in the final decision surface (Fig. 11(a) bottom-right). The DFF shown in Fig. 11(a)(top-right) guided the tracker by emphasizing the direction taken by most particles from the current location of the target. Similarly, the SFF shown Fig. 11(a)(bottom-left) allowed the tracker to consider the direction that the objects take to the exit the scene. This way, the DFF and SFF helped resolve the appearance ambiguity and allowed our tracker to maintain the correct label.

Mean-Shift Comparison: We performed a quantitative analysis by comparing the results with a color based mean-shift tracker. The comparison was performed using the ground-truth generated for the three marathon sequences. The mean-shift was initialized using the same regions and appearance was updated during the course of tracking. The tracking error was computed as the distance between the centroid of the target returned by the tracker and the corresponding centroid in the ground-truth. The error is then averaged over the entire video. The results are summarized in Fig. 12(a). The green bars in the graph correspond to the average error of our tracking algorithm, while the yellow bars correspond to the average error committed by the mean-shift tracker. The tracking error of the mean-shift is especially higher for the trajectories of Marathon-3 sequence (bars 71-85). The reason was the gradual change in the appearance of the athlete as he/she runs along the curved portion of the track. The mean-shift algorithm that only uses the appearance information was not able to cope with it and often drifted away very quickly. While, using the crowd flow and scene constraints in the form of the SFF and DFF, our algorithm was able to stay on the target for much longer time durations. In general, it can be observed from the graph (Fig. 12(a)) that our method works much better than the mean-shift tracking method. This verifies our initial observation that when tracking in videos with high density crowds appearance alone is not a reliable cue, and, other sources of information present in the scene should be exploited.

Contribution of Floor Fields: Next an experiment was performed to test the contribution of floor fields towards the accuracy of tracking. The experiment was performed using those ground-truth trajectories from Marathon-1 for which we obtained successful tracking results using all three floor fields. We picked these trajectories as it would allow us to establish whether all three floor fields contributed towards the success or not. There were 35 such trajectories in total

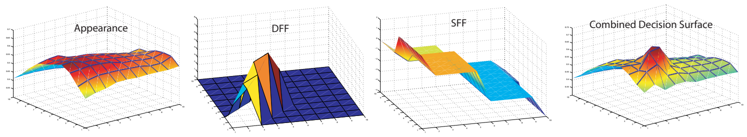


Fig. 11. (Left to Right)The appearance similarity surface, local DFF, local SFF, and the final decision surface obtained by merging appearance, the DFF, and the SFF according to 1.

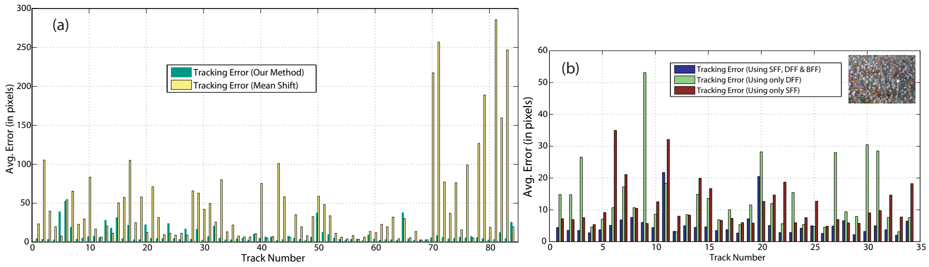


Fig. 12. (a) Comparison of the tracking error of our method against the mean-shift tracker. The bars represent the average error over the entire track. The length of the track is given in Fig. 9(1 to 50 Marathon-1; 51-70 Marathon-2; 71-85 Marathon-3). (b) Contribution made by different floor fields towards the tracking accuracy.

in Marathon-1. To measure the contribution of different floor fields, we ran our tracker first using only the SFF and then using only the DFF. The error was computed in a manner similar to that of the mean-shift experiment. The graph in Fig. 12(b) shows the comparison. It can be observed that we obtained the minimum error by using all three fields. However, the error committed by the DFF and SFF is not consistent across all the tracks. By carefully observing these 35 trajectories, we noted that SFF based tracking commits less error when the tracked athlete was running in more or less straight path with no side-ways movement, i.e. when he was perfectly obeying the geometry of the learned SFF (Fig. 6:Center) which is true for most athletes in this scene. The DFF based tracking committed relatively higher error in situations when the neighboring athletes of the target were trying to go side-ways to over-take each other, and far less error in cases when the target athlete and the neighboring athletes were running in tandem.

5 Conclusion

We have developed an algorithm for tracking targets in high density crowd scenes. The algorithm utilizes crowd flow and scene layout constraints to predict the future locations of the target. These constraints are captured by learning three floor fields, namely the Dynamic Floor Field, the Static Floor Field, and the Boundary Floor Field. These floor fields together with the appearance information is then used to track individuals in complex scenarios. The experimental results on complex videos verified that using high level knowledge about the scene in the form of floor fields is helpful in improving the tracking accuracy. The present work can take a number of future research directions: First, the individual target-tracking can be extended to multi-target tracking by treating the multi-target configuration as a multi-particle system and adding particle-particle interaction terms in the tracking framework. Second, the computation of floor fields can be improved to handle multi-modality of motion at sources, sinks, and paths in between. This will be important for handling the crowded situations where people can move in multiple directions.

Acknowledgements: This research was funded by the US Government VACE program. We would like to thank Min Hu for her valuable contribution towards this reserach.

References

1. C. Burstedde, K. Klauck, A. Schadschneider, J. Zittartz, *Simulation of Pedestrian Dynamics Using a Two-dimensional Cellular Automaton*, Physica A, 295(3), 2001.
2. A. Kirchner et al., *Simulation of Evacuation Processes Using a Bionics-Inspired Cellular Automaton Model for Pedestrian Dynamics*, Vol. 312, No. 1-2, 2002.
3. S. C. Lee and R. L. Hughes, *Exploring Trampling and Crushing in a Crowd*", J. Transp. Engrg., Vol. 131, No. 8, 2005.
4. M. Betke et al., *Tracking Large Variable Numbers of Objects in Clutter*, IEEE CVPR, 2007.
5. K. Li and T. Kanade, *Cell Population Tracking and Lineage Construction Using Multiple-Model Dynamics Filters and Spatiotemporal Optimization*, International Workshop on Microscopic Image Analysis with Applications in Biology, 2007.
6. Z.Khan, T. R. Balch, and F. Dellaert, *An MCMC-based Particle Filter for Tracking Multiple Interacting Targets*, ECCV, 2004.
7. C. Stauffer, *Estimating Tracking Sources and Sinks*, IEEE Workshop on Event Mining, 2003.
8. G. Gennari and G. D. Hager, *Probablistic Data Association Methods in Visual Tracking of Groups*, IEEE CVPR, 2004.
9. Y.Cai et al., *Robust Visual Tracking of Multiple Targets*, ECCV, 2006.
10. A. Yilmaz et al., *Object Tracking: A Survey*, ACM Journal of Computing Surveys, Vol. 38, No. 4, 2006.
11. S. Ali and M. Shah, *A Lagrangian Particle Dynamics Approach for Crowd Flow Segmentation and Stability Analysis*, IEEE CVPR, 2007.
12. G. Brostow and R. Cipolla, *Unsupervised Bayesian Detection of Independent Motion in Crowds*, IEEE CVPR, 2006.
13. T. Zhao and R. Nevatia, *Bayesian Human Segmentation in Crowded Situations*, IEEE CVPR, 2003.
14. T. Zhao and R. Nevatia, *Tracking Multiple Humans in Crowded Environment*, IEEE CVPR, 2004.
15. R. Gurka et al., *Computation of Pressure Distribution Using PIV Velocity Data*, In Proc. of the 3rd International Workshop on Particle Image Velocimetry, 1999.
16. D. Comaniciu and P. Meer, *Mean Shift: A Robust Approach Toward Feature Space Analysis*, IEEE TPAMI, 2002.