A Multiple Targets Appearance Tracker Based on Object Interaction Models

Guorong Li, Wei Qu, *Member, IEEE,* and Qingming Huang, *Senior Member, IEEE*

***Abstract*—Kernel-based method has been proved to be effective in solving single-target tracking problem. However, facing more complicated multitarget tracking task, most classic kernel-based multitarget trackers do not model the interaction among targets successfully and simply track each target independently. Thus, they usually could not deal with "singularity" problem and fail in tracking the target when occlusions occur or distracters appear. Although multikernel methods may improve the performance by introducing more constraints, how to simulate the relationship and interaction among the tracked targets is still not fully investigated. In this paper, we discuss a very common scenario of multitarget tracking, in which a moving object's motion is not only determined by its virtual destination but also impacted by other neighboring objects. This phenomenon exists in many usual tracking applications such as human tracking, traffic monitoring, video surveillance, and so on, where an object usually moves toward a particular direction but meanwhile detours when close to others to avoid collision. Specifically, we propose a novel interaction model to explain the above phenomenon. Then by defining a new cost function, we embed this interaction model into a kernel-based tracker and further derive our interactive kernel-based multitarget tracker. Experimental results on various datasets demonstrate that our interaction model can alleviate "singularity" problem and, thus, the proposed tracking method could achieve superior performance in multitarget tracking.**

***Index Terms*—Interaction model, Kalman filter, kernel tracking, multitarget tracking, virtual destination, virtual gravity, virtual repulsion.**

I. Introduction

MULTITARGET trackers have achieved efficient and robust performance in many challenging scenarios as reported in [1]–[9]. Okuma *et al.* first proposed a combination of AdaBoost for object detection with particle filter for multitarget tracking in [4]. This combination addresses both detection and consistent track information in the same framework and leads to fewer failures than either one on its own. The work of [6] combines detection algorithm with particle filter to tackle abrupt motions of moving objects in low frame rate video. Meanwhile many works, such as [7]–[9], formulate multitarget tracking into the data association problem based on detection results. Recently, more and more literatures [10]–[16] begin to pay attention to scene information and interactions between targets. In [10], the authors assumed the interaction performs like pairwise Markov random field (MRF). They proposed a full joint state-space model to track fixed number of interacting ants efficiently, by sampling from the target posterior distribution in particle filter using MCMC sampling. While in [11], dynamic spatial patterns inside the targets are represented by using a mixture of MRF. Local and global relations among multiple targets and preferences are encoded into pairwise potential functions of MRF. To track targets in high density crowd, Ali and Shah [12] utilized crowd flow and scene layout constraints to predict future state of the target by learning static floor field, dynamic floor field, and boundary floor field. While in [13], linear trajectory avoidance (LTA) model of dynamic social behavior is proposed. It needs to be trained off-line with videos taken from top-view angle. Then it can be applied for predicting the states of targets. As the environment in surveillance applications is usually fixed, in order to make full use of structured environment dynamic information, the work [14] incorporates the environment state into the target's state-space in particle filter and uses rejection sampling to help generate more effective particles. Besides, since no tracker is perfect for object tracking in all cases, in order to switch between trackers, [15] and [16] defined local spheres of influenced, based on which the interaction level of the tracked targets can be decided.

However, most of the above methods only present tracking results to demonstrate their methods with little theoretical analysis. Besides, few effective interaction models have been successfully combined with kernel-based tracker [17] which usually suffers from "singularity" problem [18], in which when observation is given, the state of the target cannot be determined uniquely. Even though the earlier efforts [19]–[21] were made to tackle this problem through imposing constraints on relationships among different kernels or selecting motion models, there still exist many open problems especially for multitarget tracking. For example, how to select kernel? How many kernels or motion models should be used? One of the most important problems is how to model the interaction among multiple objects for kernel-based tracker. This has been partially investigated for articulated object tracking [20], where

Manuscript received April 27, 2010; revised June 28, 2010, September 9, 2010, October 17, 2010, January 17, 2011, and April 8, 2011; accepted July 21, 2011. Date of publication August 22, 2011; date of current version March 7, 2012. This work was supported in part by the National Natural Science Foundation of China, under Grants 61025011 and 60833006, in part by the National Basic Research Program of China (973 Program), under Grant 2009CB320906, in part by the Beijing Natural Science Foundation, under Grants 4111003 and 4092042, in part by the Open Projects Program of the National Laboratory of Pattern Recognition, under Grant 20090021, and in part by the President Fund of GUCAS. This paper was recommended by Associate Editor H. Gharavi.

The authors are with the Chinese Academy of Sciences, Beijing 100080, China (e-mail: grli@jdl.ac.cn; qu.wei@yahoo.cn; qmhuang@jdl.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2011.2165591

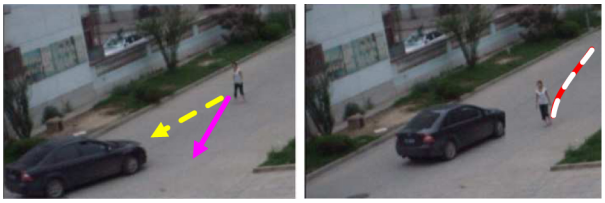


Fig. 1. Common scenario of multitarget tracking. A girl and a car are moving toward each other and the yellow arrow denotes her original velocity. As a result, both the girl and the car would take actions to avoid collision. The pink arrow represents her actual velocity. The red and white line in the right image is her trajectory.

the interaction among different tracking parts could be known by some prior knowledge. However, for general multiple target tracking, this problem is much more difficult since it is not easy to discover the relationship among the tracked objects. This is why few successful kernel-based multitarget tracking methods have been reported. To the best of our knowledge, kernel-based multitarget tracking with effective interaction model has not been thoroughly studied.

With the above problem in mind, what is the interaction we are looking for? Fig. 1 illustrates an example, where a girl walks in the street and a car is coming close to her. In such a scenario, the girl will normally turn around to avoid the oncoming car instead of continuing walking forward as shown in the right image. Nonetheless, if there was not this car, typically, a person would keep his moving direction since the road is straight. From this example, we can easily perceive the interaction happening between the walking girl and the oncoming car. Such kind of situation exists in many places such as street scene, highway road, union station, airport, and so on, where an object's motion and future movement are affected by its surroundings such as other moving targets, obstacles, and so on. In this paper, we investigate this observation and propose a novel interaction model to simulate such kind of objects' behavior in the above scenario. Then, we present a new kernel-based multitarget tracking method based on this model and it achieves superior performance compared with kernel-based tracker without interaction model.

Here, we want to stress that although both our model and LTA [13] have something in common with social force model [22], they still have three crucial differences. First, in [13], a homography matrix is needed to transform image plane to world coordinates which requires users manually click points. This severely restricts its utility. Second, the idea in [13] needs users to label goal points at the beginning while we automatically discover our virtual destinations. Third, LTA assumes that every pedestrian knows the states of all objects and they will move at a constant velocity which is not true in the reality. In our model, we suppose that acceleration of an object (pedestrian or vehicle) varies uniformly and presume the object is aware only of neighbors within some specific area and keeps paying attention to them when it moves.

The rest of this paper is organized as follows. Section II presents the interaction model we propose. Specifically, we discuss two different kinds of interaction in Sections II-A and II-B, respectively. Then in Section II-C, the relationship between the tracked target's interaction and motion is derived

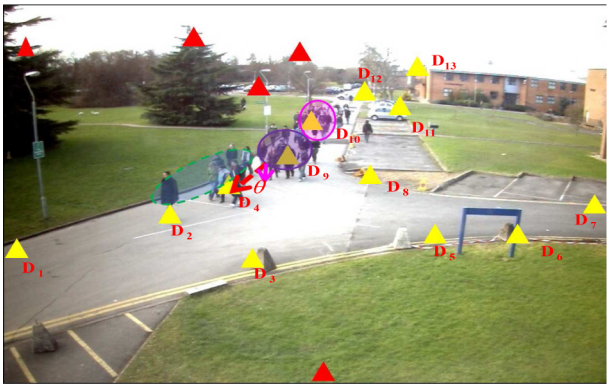


Fig. 2. Triangles are candidate VDs obtained automatically. Red ones are discarded during tracking. Pink and red arrows denote velocity vector and distance vector, respectively. θ is the angular displacement between them.

and used to estimate its future position. In Section III, we present a novel kernel-based multitarget tracking framework by providing a new cost function and exploiting the proposed interaction model. Experimental results are reported in Section IV. Finally, we conclude this paper in Section V.

II. Our Interaction Model

Among many common tracking applications such as vehicle tracking in traffic monitoring and pedestrian tracking in public video surveillance, we observe an interesting phenomenon that the tracked objects often tend to keep their moving direction without any outside interaction. For instance, cars normally follow traffic lanes and pedestrians walk along streets. In other words, it seems that these targets have intentions to go to some destination during a particular period. This is commonly true in reality. For example, a car would head to next traffic light or road crossing. A pedestrian's movement may be more complicated, but usually he would also have similar consciousness when walking. Maybe there exist some exceptions, however, this observation is valid in a lot of tracking scenarios. Therefore, it could be exploited to improve tracking performance especially when many targets appear simultaneously in the scene. On the other hand, we also observe that moving objects normally do not intend to collide with the others no matter how fast they move in the above discussed applications. They can adjust their motion to avoid collisions or keep a distance away from others while still trying to reach their destinations as soon as possible. Fig. 2 shows an example, where lots of pedestrians are walking in the scene. It can be seen that they intend to move toward several certain directions, yet no one collides with others. In this section, we thoroughly investigate the above phenomenon and propose two different interaction models: one is about the interaction between a tracked target and its virtual destination; the other simulates the interaction between one tracked target and other neighboring targets.

A. Interaction Between A Moving Object and Its Virtual Destination

Although a tracked target's real destination is usually unknown, some points where the target will probably arrive in

(a) (b)

Fig. 3. Example illustrating places of potential VDs: moving object, image sides, and edges are likely to be the candidates for VDs. (a) A, B, C and the green region are potential VDs. (b) P_1, P_2, P_3 and P_4 are potential VDs.

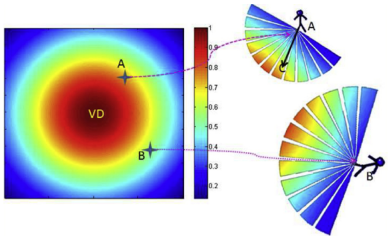


Fig. 4. Left image shows VG field (the first item of 1) of a VD. Points A, B represent positions of the tracked targets. The right two semi-circle images denote the strength of VG that VD attracts A and B, respectively. For example, the vector (black arrow) connecting person A and point C denotes his velocity while the color of point C denotes the strength of VG that is exerted on A if he moves with that velocity.

a short period could still be easily discovered and designated. Since those points are assumed to be the destinations of the moving objects and they may be not very accurate, we define those points as virtual destination (VD). Algorithms for finding sinks in [12] and [23] can be used for discovering VDs, because sinks are the last states (stop states) of particles and thus can been considered as VDs. However, those algorithms are more suitable for scenes with crowds and their procedures are a little complicated. Therefore, we propose a simple but effective method to discover potential VDs automatically.

First, in a fixed environment, objects' motions usually share common characteristics because of the constraints of the environment structure. So a moving target's position could become the future position of another object. Fig. 3(a) shows an example, cars drive along the street and apparently the position of car A would be the future position of car B. So we can assume that the position of car A is the VD of car B. Second, places where the moving objects may stop or enter could be potential VDs. Since edges usually exist where two different kinds of regions (e.g., road and grass, road and house) are adjacent, we could consider that some points [e.g., red points $P_1 \cdots P_4$ in Fig. 3(b)] on those edges has high probability to become VDs. Finally, the image plane can only cover a limited zone, so the four sides of the image [e.g., green region in Fig. 3(a)] are also the probable places where the object enters or disappears. Thus, points on those sides are also likely to be VDs. For example, the green region in Fig. 3(a) is the destination of car C. Considering the above analysis, in our experiment, we use optical flow to locate potential VDs. We first find out good feature points, which usually locate on edges and moving objects, and then compute their optical flow. As we can see, feature points are not sparse enough. In order to reduce the storage space and computational complexity of the following procedure, we cluster them according to their spatial positions and optical flow. Clustering centers are initialized as potential VDs (e.g., red and yellow triangles in Fig. 2). Based on the third analysis, if there is not any potential VDs located on the sides of the image, we place several VDs on them uniformly. Then during the tracking process, the pre-selected VD in front of a particular target would be selected as its current VD. Moreover, those VDs (e.g., red triangles in Fig. 2) which are not used for a long time are discarded. In Fig. 2, D_4 is selected as the VD of the crowds covered by purple ellipse and apparently, there is a high probability that they will reach D_4.

As objects usually intend to arrive at their VDs, we could naturally assume that VDs are attractive to those associated objects. Since this kind of interaction is very similar to gravity in physics, we define the attraction force of a VD as its virtual gravity (VG). It is easy to understand that different targets may have different VDs and thus enjoy different VGs. However, two confusing cases need to be clarified. First, VGs associated to different VDs of the same tracked target may not be equivalent. For example, if a person walks faster toward one of his VDs, it means that this VD is more attractive to him and thus the associated VG would be stronger. Second, different targets sharing the same VD may also have different VGs. For instance, if a person runs very fast, the possibility that he changes direction or stops is much lower than that of those people who walk slowly or wander around aimlessly. In this case, we say that the VG associated to this person is bigger. Furthermore, it is usually abnormal for a car or pedestrian that moves with high speed to make a sharp turn or stop immediately which is very dangerous. In these cases, the VG is designed stronger so that it would prevent the object from changing direction significantly or stopping even though they may collide with the others (e.g., some traffic accidents).

Besides the velocity, distance is another factor that affects VG. We define VG as inversely proportional to the distance between tracking object and its associated VD. This definition shares similarities to the gravity force in physics. It can also be easily verified in many common scenes. For instance, when an object is far away from its VD, it has more time and flexibility to vary its motion. In other words, its VD has less control on this object. Thus, the VG is small. However, when this object is getting closer to its VD, such kind of attraction becomes increasingly stronger and thus the object has less time and flexibility to make any changes of its motion before reaching the targeting VD. We could say that in this case the VG is stronger.

Considering the two factors mentioned above, we simulate the interaction between an object and its VD with a gravity model (1) whose first item shares similarities with the gravity force in physics

$$VG(x_t^i, g_t^i) \propto \underbrace{\frac{\lambda_g}{\| x_t^i - g_t^i \|^2}}_{\text{first item}} \exp\left(\frac{\| v_t^i \cos(\theta) \|^2}{\sigma_v^2}\right) \underbrace{\frac{g_t^i - x_t^i}{\| x_t^i - g_t^i \|}}_{\text{direction of VG}} \tag{1}$$

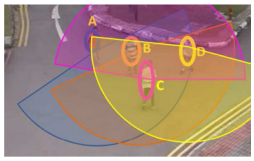


Fig. 5. Blue, orange, pink, and yellow fan-shape areas are RA of person A, RA of B, RA of C, and RA of D, respectively.

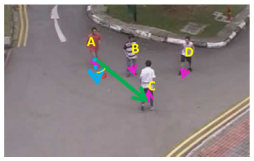


Fig. 6. Four persons walk with different velocities (denoted by pink arrows). We take person C for example and illustrate the effects of person A, B, and D.

where N_t is the number of the tracked targets at time t,[1] λ_g is the gravity constant, x_t^i, v_t^i, and g_t^i represent ith object's state, velocity (pink arrow in Fig. 2), and VD at frame t, respectively. θ (red θ in Fig. 2) is the angular displacement between v_t^i and g_t^i. The first item in (1) is referred to as VG field. Fig. 4 shows an example of VG field of a VD and its VG toward objects.

B. Interaction Between A Moving Object and Other Neighboring Objects

When an object is moving, it usually does not ignore its surroundings. For example, if there is a stranger close to a walking person, he would be normally aware of this stranger's motion. When watching a person walking toward him, what would this person do? Change moving direction to avoid the oncoming collisions or just continue to walk with the original velocity. Generally, most people will choose the former action. However, intuitively a person could not notice all of his surroundings because of his eyesight and some other factors. Research about inter personal space distance and the corresponding actions in anthropology pointed out that within a certain distance (called "public distance" in [24]), people would become aware of objects in front and begin to take actions to avoid the oncoming collisions with them. In other words, if other people enters some psychological areas around a person's physical body, the person may feel anxious. In other words, only objects within this area have influences on that person. This is very similar to cross-section of reconnaissance radar within which objects can be detected by radar. Therefore, the term reconnaissance area (RA) of a person could be defined as the space around him, where if other objects enter, he will start to pay attention to their motions. As in common scenes, people usually only care areas in front, so we limit RA as a fan-shaped area as shown in Fig. 5, where four pedestrians are the tracked targets and their RAs are labeled with different colors. It can be seen that B is in A's RA while A is not in B's. In this paper, we simply assume that object A has a direct effect on object B if and only if A is in the RA of B. So the interactions among moving objects are asymmetrical. For simplicity, we define interactive neighbor (IN) of an object as the neighbor that has direct effect on it.

In most cases, if B is in A's RA, A will take actions to avoid colliding with B or keep a certain distance from it. It seems that B has a kind of repulsion force that prevents A from getting closer to it. Borrowing the concept of repulsion force in physics, we refer the effect of an object to another as its virtual repulsion (VR) force. Intuitively, VR between different objects is not the same. For example, a fast moving oncoming car is more dangerous than a person who walks away. Therefore, the car has stronger VR. Fig. 5 presents another example, where both people B and C are INs of person A. However, apparently C is more dangerous to person A because she walks toward him. We also notice that even though B is closer to A than C, B walks away from person A and is not dangerous at all. As a result, person A will first take actions to avoid colliding with C. It could be explained that C has stronger VR toward A than B. We can also say that C has stronger VR than B and this force becomes the dominant VR that compels A to change his moving direction and speed. Meanwhile, although D and A have the same velocity, C will pay more attention to D because D is closer than A. So it is reasonable to say that for C, VR of D is stronger than that of A. According to previous research [24], there exist some other objective factors, such as gender, age, eyesight, which affect the calculation of VR in reality. But according to the state-of-the-art technology in computer vision, it is difficult to obtain these objective factors accurately without prior knowledge. Here for generality, we only consider two subjective factors: distance and velocity, and model the VR between two objects in the following way.

Let $v_t^{i_k}$ denote the velocity of kth IN of the target x_t^i at time t, ϑ represent angular displacement between relative position vector $x_t^i - x_t^{i_k}$ and relative velocity vector $v_t^{i_k} - v_t^i$. As shown in Fig. 6, the blue arrow is the relative position vector of C to A and the blue dotted arrow is the relative velocity vector of A to C. Angular displacement between the two vectors is labeled as red ϑ. Then we can derive the smallest distance $P(x_t^i, x_t^{i_k})$ between the two objects if they do not take any changes and the time $T(x_t^i, x_t^{i_k})$ that will take them to reach the closest position

$$T(x_t^i, x_t^{i_k}) = \| x_t^i - x_t^{i_k} \| / (\| v_t^{i_k} - v_t^i \| \cos\vartheta) \tag{2}$$

$$P(x_t^i, x_t^{i_k}) = \| x_t^i - x_t^{i_k} \| \, ctan\vartheta. \tag{3}$$

Combining the above analysis, repulsion model between an object x_t^i and its IN $x_t^{i_k}$ is formulated as

$$VR(x_t^i, x_t^{i_k}) \propto \exp\left(\frac{-P^2(x_t^i, x_t^{i_k})}{\sigma_p^2}\right) \exp\left(\frac{-T^2(x_t^i, x_t^{i_k})}{\sigma_t^2}\right) \underbrace{\frac{x_t^i - x_t^{i_k}}{\| x_t^i - x_t^{i_k} \|}}_{\text{direction of VR}}, k = 1, 2, \cdots, N_t^i \tag{4}$$

where N_t^i is the number of INs of x_t^i.

[1]Here, for simplicity, we can consider t as the time of tth frame. So we need not consider the frame interval.

C. Relationships Between the Tracked Object's Interaction and Motion

As discussed above, many virtual forces are applied on a moving object, so what are their effects and how would object's motion change? These questions are obviously very important to object tracking problem. Forces and motions are classical topics studied in physics. Newton's second law of motion states that the net external force on a body is equal to the mass of that body times by its acceleration. It allows quantitative calculations of dynamics: how does velocity change when forces are exerted on an object. Borrowing these theories, we use (5) and (6) to model the relationships among the object's motion, and VGs, VRs

$$f_t^i(x_t^i, v_t^i) \models VG(x_t^i, g_t^i) + \sum_{k=1}^{N_i^t} VR(x_t^i, x_t^{i_k}) \tag{5}$$

$$a_t^i \propto f_t^i(x_t^i, v_t^i). \tag{6}$$

As the VG and VR are always changing especially in low-frame rate videos, acceleration of the object varies all the time. So we assume that each object moves with uniform changes in acceleration from previous frame to current frame instead of moving with a constant acceleration. Then according to the accelerations at time $t-1$ and t, applying equations of straight line in 3-D space, we can derive the acceleration at any time between $t-1$ and t using (7). By applying the connections between position, velocity and acceleration [25], we can further derive the velocity and position of the object at any time between $t-1$ and t

$$a_{t-1+\lambda}^i = a_{t-1}^i + \lambda(a_t^i - a_{t-1}^i), 0 \le \lambda \le 1 \tag{7}$$

$$v_{t-1+\lambda}^i = v_{t-1}^i + \int_{t-1}^{t-1+\lambda} a_\tau^i d\tau \tag{8}$$

$$= v_{t-1}^i + \int_0^{\lambda} a_{t-1+\lambda}^i d\lambda \tag{9}$$

$$= v_{t-1}^i + a_{t-1}^i\lambda + \frac{1}{2}(a_t^i - a_{t-1}^i)\lambda^2 \tag{10}$$

$$x_{t-1+\lambda}^i = x_{t-1}^i + \int_{t-1}^{t-1+\lambda} a_\tau^i d\tau \tag{11}$$

$$= x_{t-1}^i + \int_0^{\lambda} v_{t-1+\lambda}^i d\lambda \tag{12}$$

$$= x_{t-1}^i + v_{t-1}^i\lambda + \frac{1}{2}a_{t-1}^i\lambda^2 + \frac{1}{3}(a_t^i - a_{t-1}^i)\lambda^3. \tag{13}$$

From (8) to (9), we introduce a new variable $\lambda = \tau - (t-1)$ and replace τ with it. Then substituting into (7) and applying integral operator, we can obtain (10). The processes of derivation from (11) to (13) are similar and we do not repeat the explanation any more. According to (10) and (13), we can get v_t^i and x_t^i by setting $\lambda = 1$. That is

$$v_t^i = v_{t-1}^i + \frac{a_t^i + a_{t-1}^i}{2} \tag{14}$$

$$x_t^i = x_{t-1}^i + v_{t-1}^i + \frac{a_t^i + 2a_{t-1}^i}{6}. \tag{15}$$

To simplify the expression of this model, we define

$$\psi(x_t^i, v_t^i) \models v_{t-1}^i + \frac{a_t^i + a_{t-1}^i}{2} \tag{16}$$

$$\phi(x_t^i, v_t^i) \models x_{t-1}^i + v_{t-1}^i + \frac{a_t^i + 2a_{t-1}^i}{6} \tag{17}$$

and then the interaction model can be written as follows:

$$v_t^i = \psi(x_t^i, v_t^i) \tag{18}$$

$$x_t^i = \phi(x_t^i, v_t^i). \tag{19}$$

To demonstrate the efficiency of our interaction model, we use it to predict moving objects' future positions in videos whose frame rates are down sampled into 5 f/s. Fig. 7 shows some results and we can see that the proposed VG and VR are able to estimate the actual position.

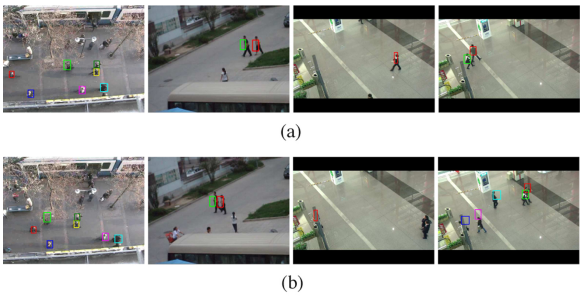


Fig. 7. Prediction results on low frame rate videos using the proposed interaction model. Predicted positions of moving objects are presented by rectangles. (a) Prediction results of our interaction model. (b) Prediction results after several frames. We can see that in the third image of (b), the person is occluded completely, but our interaction model could estimate his position accurately.

III. Kernel-Based Multitarget Tracking Using the Proposed Interaction Models

In classic kernel-based methods, object tracking is often formulated as an optimization problem

$$\hat{x}_t = \text{argmin}_{x_t} \parallel \varphi(x_0) - \varphi(x_t) \parallel^2 \tag{20}$$

where $\varphi(x_0)$ and $\varphi(\cdot)$ are functions of an image patch: $\varphi(x_0)$ is the prior model of the tracked target and $\varphi(x_t)$ is the observation of candidate region. For multitarget tracking cases, the optimization problem becomes

$$\hat{x}_t^i = \text{argmin}_{x_t^i} \parallel \varphi(x_0^i) - \varphi(x_t^i) \parallel^2, i = 1, 2, ..., N_t. \tag{21}$$

Unfortunately, no target interaction is taken into account in (21) due to the difficulties of analyzing relationship among the

tracked targets. We propose to integrate our interaction models as constraints on kernel-based multitarget tracking. The new cost function is as follows:

$$\hat{x}_t^i = \text{argmin}_{x_t^i} \parallel \varphi(x_0^i) - \varphi(x_t^i) \parallel^2 \tag{22}$$

$$\text{subject to } x_t^i = \phi(x_t^i, v_t^i) \tag{23}$$

$$v_t^i = \psi(x_t^i, v_t^i). \tag{24}$$

By applying the penalty function method [26], the above problem is also equivalent to minimizing the following cost function:

$$\underbrace{\parallel \varphi(x_0^i) - \varphi(x_t^i) \parallel^2}_{\text{first item}} + \underbrace{\lambda_1 \parallel \psi(x_t^i, v_t^i) - v_t^i \parallel^2}_{\text{second item}} + \underbrace{\lambda_2 \parallel \phi(x_t^i, v_t^i) - x_t^i \parallel^2}_{\text{third item}} \tag{25}$$

where λ_1 and λ_2 are positive penalty coefficients. The second item denotes the punishment for velocity deviation from our interaction model and the third item represents the punishment for position deviation. Intuitively, by introducing two proposed interaction models, our trackers could become more intelligent since they know how to keep moving toward different VDs as well as taking action to avoid collisions simultaneously. Therefore, the multitarget tracking performance could be greatly improved. For simplicity, we refer our derived tracker as interactive kernel-based multitarget tracker (IKMT). In the following subsections, we further analyze the advantages of the proposed IKMT theoretically.

A. Observability Analysis

After applying Taylor expansion, the solution of (25) is the same as linear system (the detailed derivation is provided in Appendix A in supplementary material)

$$\Delta x_t^i = A_t^i \Delta x_{t-1}^i + B_t^i(\Delta v_t^i + u_t^i) + \omega_{t-1}^i \tag{26}$$

$$\Delta v_t^i = M_t^i \Delta v_{t-1}^i + D_t^i \Delta x_t^i + l_t^i + \varepsilon_{t-1}^i \tag{27}$$

$$z_t^i = C_t^i \Delta x_t^i + \xi_{t-1}^i \tag{28}$$

where $\Delta x_t^i = x_t^i - x_0^i$, $\Delta v_t^i = v_t^i - v_0^i$, B_t^i, A_t^i, u_t^i, M_t^i, D_t^i, l_t^i and C_t^i are matrix variables defined by (A.20)–(A.22), (A.15)–(A.17), and (A.7), ω_t^i, ε_t^i and ξ_t^i are noise terms. It can be seen that without exploiting interaction models, the proposed method could degenerate into the classic kernel-based tracker [19], which only has observation equation (28).

Similar to the work in [21], we use observability theorem [27] for analysis.

Observability Theorem: A system is observable if and only if its observability matrix has full rank, where if C_t and A_t are observation matrix and state transition matrix, $O_t = [C_t \quad C_t A_t \quad \cdots \quad C_t A_t^{n-1}]^T$ and $O_t = C_t$ if there is no state transition matrix. The observability matrix O_t^i of the system (26)–(28) is as follows:

$$O_t^i = \begin{pmatrix} C_t^i \\ C_t^i A_{t-1}^i \\ \vdots \\ C_t^i \prod_{k=0}^{t-1} A_k^i \end{pmatrix}. \tag{29}$$

We would like to compare the rank of this matrix with two types of existing kernel-based tracking methods and thus show why the proposed IKMT is superior to [17], [20], and [21].

1) As analyzed in [21], many classic kernel-based tracking methods [28] could be unified into one observation equation as shown in (28). In this case, the observability is $rank(C_t^i)$. Since $rank(O_t^i) \geq rank(C_t^i)$, our IKMT's observability is not less than that of classic kernel-based tracker. In other words, the proposed method has the ability to increase tracking observability due to introducing more effective interaction models. Therefore, it could achieve much better tracking performance as demonstrated by the experimental results.
2) Several state transition matrices [20], [21] have been used to improve the kernel-based tracking method. As discussed in [21], some of them use Tikhonov regularization [29]. For example, one common selection of the motion model is

$$x_t^i = G_t x_{t-1}^i. \tag{30}$$

In this case, the observability matrix is as follows:

$$\widetilde{O}_t^i = \begin{pmatrix} C_t^i \\ C_t^i G_{t-1}^i \\ \vdots \\ C_t^i \prod_{k=0}^{t-1} G_k^i \end{pmatrix}. \tag{31}$$

Simply comparing (29) to (31) at first glance, it may be hard to see the difference in mathematics. However, the improvements lie in the constraints. In (31), the matrix G comes from the motion model (30). As discussed in [21], usually it is not easy to get an accurate estimation of object's motion matrix without prior knowledge, especially for multiple object tracking applications. This is why most previous papers adopt simple motion matrices such as identity matrix for simplicity. Apparently, it is not enough. Nevertheless, the proposed interaction models in Sections II-A and II-B provide a much better description for object's motion by not only considering object's virtual destination but also the interaction with its neighbors. Thus, our IKMT has more potential to increase the observability than (31). In other words, $rank(O_t^i)$ is normally higher than $rank(\tilde{O}_t^i)$. Thus, the tracking performance is more robust accordingly as shown in the experimental results.

B. Iteratively Distributed Algorithm for Multitarget Tracking

Since each tracking object is only affected by its VD and neighboring objects in its RA, we regard each tracking

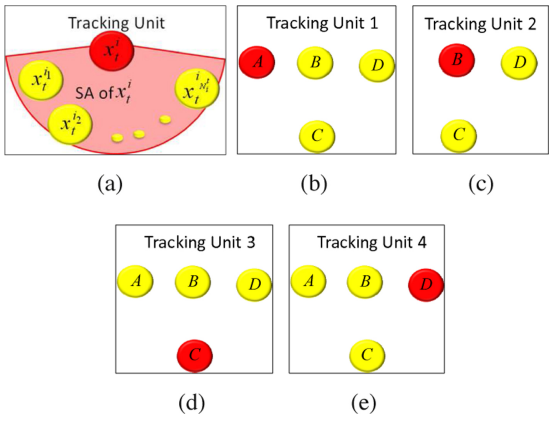


Fig. 8. (a) Tracking unit for ith target. (b)–(e) Tracking units that are decomposed from Fig. 5 for objects A, B, C, and D, respectively.

object, its VD and INs as one tracking unit. Fig. 8 shows an example, where each target i belongs to a tracking unit. With this scheme, multitarget tracking is converted to track all these units, respectively. For instance, the four people in Fig. 5 generate four tracking units (b), (c), (d), and (e) as illustrated in Fig. 8. Each tracking unit could be represented by (26), (27), and (28), where (26) and (28) construct a linear control system by regarding $v_t^i + u_t^i$, x_t^i, z_t^i as input, state, and observation, respectively. Different methods in control theory could be applied to solve the above system. In this paper, we adopt the recursive Kalman method [27] to show an example. Table I provides the detailed procedure of our algorithm. Because v_t^i and x_t^i are restricted by each other, we first initialize v_t^i and then perform an iterative procedure to update x_t^i and v_t^i. After that, in order to get a steady solution, we propose another iterative procedure as shown in Table II to estimate the positions of multiple tracked targets since they have interaction and dynamically affect each other.

TABLE I
RECURSIVE PROCEDURES OF OUR SOLVING ALGORITHM

	Assume that we have obtained velocity, position of objects except for object i
Step 1:	Initialize $\Delta v_t^i = \Delta v_{t-1}^i$, $\Delta \widehat{v}_t^i = \Delta v_{t-1}^i$ and Threshold T_1
Step 2:	Estimate Δx_t^i
Step 2.1:	Compute B_t,A_{t-1} and $(\Delta v_t^i + u_t^i)$ according to equations
Step 2.2:	Time update
Step 2.2.1:	Predict current state according to the previous state $\Delta \widehat{v}_t^i = A_T^i \Delta x_{t-1}^i + B_t^i \Delta v_t^i + B_t^i u_t^i$
Step 2.2.2:	Update prior error covariance [18] $P_t^i = A_t^i S_{t-1}^i A_t^i + I$
Step 2.3:	Measurement update
Step 2.3.1:	Update the Kalman gain $G_t^i = P_t^i (C_t^i)^T (C_t^i P_t^i (C_t^i)^T + I)^{-1}$
Step 2.3.2:	Update estimation of current state according to current observation $\Delta x_t^i = \Delta \hat{x}_t^i + G_t^i (z_t^i - C_t^i \Delta \hat{(x)}_t^i)$
Step 2.3.3:	Update posterior error covariance [2] $S_t^i = (I - G_t^i C_t^i) P_t^i$
Step 3:	Update Δv_t^i $\Delta \widehat{v}_t^i = M_t^i \Delta v_{t-1}^i + D_t^i \Delta x_t^i + l_t^i + \varepsilon_{t-1}^i$ if $\| \Delta v_t^i - \Delta \widehat{v}_t^i \| < T_1$ or the times of iteration ≥ 10 Stop; Else $\Delta v_t^i = \Delta \widehat{v}_t^i$ Goto Step 2; End if

TABLE II
ITERATIVE CALCULATION OF THE TRACKED TARGETS' POSITIONS

Assume that we have obtained velocity, position of objects except for object i
Step 1: initialize the threshold T_2 $x_t^i = x_{t-1}^i, \widehat{x}_t^i = x_{t-1}^i, i = 1, 2, ..., N_t$
Step 2: calculate Δx_t^i, Δv_t^i according to Table I ($i = 1, 2, ..., N_t$)
Step 3: if $\sum_{i=1}^{N_t} \| \widehat{x}_t^i - x_t^i \| < T_2$ Stop; Else $x_t^i = \widehat{x}_t^i$, $i = 1, 2, ..., N_t$ Goto Step 2; End if

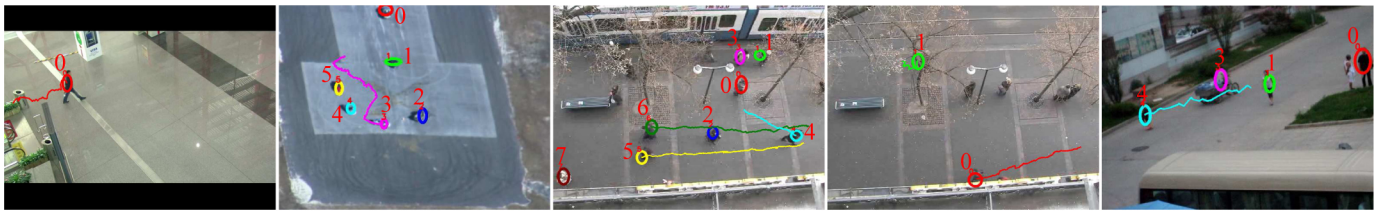
IV. EXPERIMENTAL RESULTS AND ANALYSIS

The proposed tracker is implemented in C++ code runs on 3.4 GHz Pentium IV PC without code optimization and is designed for tracking targets in the video which is recorded nearly from top-view. The initial position of every target is manually labeled with a rectangle and the $B-G$, $B-R$, and $B+G+R$ (quantified into $8\times8\times4$ bins) histogram of the pixels in target's initialized region is used as its appearance model and it is not updated during tracking. For similarity measure, there are many choices such as Matusita distance, Euclidean distance, and so on. In our experiment, Bhatacharyya coefficient is used for comparing two normalized histograms. For the value of parameters, we empirically select the radius (r) of RA as $r = \lambda \times \max(width, height)$, $\lambda \in [3, 5]$ and $\sigma_d = \sigma_p = r$, $\sigma_v \in [10, 20]$, $\sigma_t \in [2, 5]$, $\lambda_g = 10$, $T_1 = 30$, $T_2 = 10$. Without considering initialization time, the average tracking speed on the tested videos could achieve more than 20 f/s satisfying the requirement of real-time.

A. Analysis of the Effectiveness of VG and VR

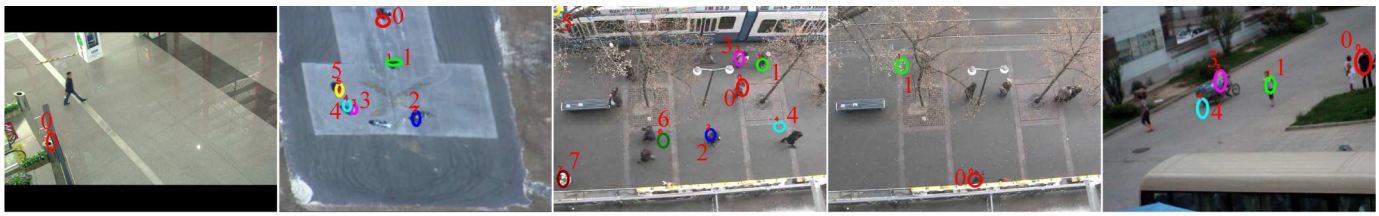
In order to test the two proposed interaction models, respectively, we implement another two trackers: IKMT only considering VG [referred to as VG tracker (VGT)] and IKMT only considering VR [referred to as VR tracker (VRT)] and compare them with kernel-based multitarget tracker (KMT) derived from [17].

1) *Analysis of the Effectiveness of VG:* Fig. 9 provides some examples for illustrating the effect of VG. Intuitively, VG attracts every target move to its associated VD and thus it could prevent the target from staying behind or drifting to the surroundings. In Fig. 10(a), a man passed by a pillar and unfortunately KMT drifts to the pillar as shown in Fig. 10(a). However, as he has VD and there are no other neighbors affecting him, according to our VG model, he would keep walking toward his VD instead of turning aside. Therefore, VGT could track it successfully and Fig. 9(a) provides tracking result of VGT. From examples shown in images Fig. 9(a)–(d), we could infer that when target's appearance is similar to background or other objects, it is nearly impossible to track the targets successfully with KMT. Moreover, Fig. 10(e) comes from a low-frame rate video (12 f/s) and target 4 (the woman labeled with cyan ellipse) runs fast. So sometimes there is no overlap between her positions in the two



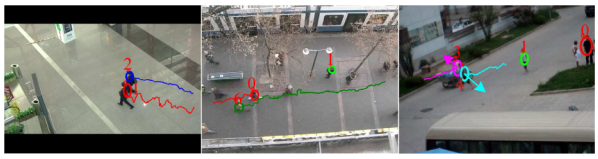
(a) (b) (c) (d) (e)

Fig. 9. Some tracking results on test videos using VGT. Trajectories of some interesting targets are represented by curves of the corresponding colors. (a) Frame 677 in "airport." (b) Frame 210 in "egtest01." (c) Frame 9165 in "seq_hotel." (d) Frame 14 745 in "seq_hotel." (e) Frame 424 in "low-frame rate video."



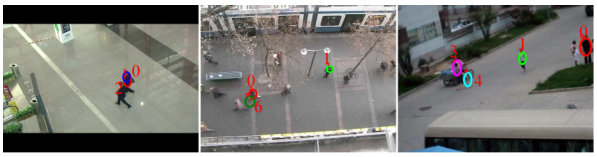
(a) (b) (c) (d) (e)

Fig. 10. Some tracking results on test videos using KMT. (a)–(e) correspond to Fig. 9(a)–(e), respectively.



(a) (b) (c)

Fig. 11. Some tracking results on test videos using VRT. Trajectories of some interesting targets are represented by curves of the corresponding colors. (a) Frame 394 in "airport." (b) Frame 8293 in "seq_hotel." (c) Frame 413 in "low-frame rate video."



(a) (b) (c)

Fig. 12. Some tracking results on test videos using KMT. (a), (b), and (c) correspond to Fig. 11(a), (b), and (c), respectively.

consecutive frames and KMT fails in tracking her. As shown in Fig. 9(e), VGT could handle this situation.

2) *Analysis of the Effectiveness of VR:* Our intention of proposing the VR model is to prevent the tracker from confusing targets when they are getting close. Fig. 11 shows some representative examples of the tracking results using VRT. From Fig. 12(a) and (b), we can see that VR could help our tracker effectively by not confusing two neighboring targets. In Fig. 12(a), two people walk toward the same direction and KMT mistakes target 2 for target 0 because their appearances are similar. However, according to our VR model, target 2 has VR toward target 0, so tracking result of target 0 using VRT would not drift to target 2 as shown in Fig. 12(a). In Fig. 11(c), target 3 and target 4 move toward each other and thus have VR on each other. So, VRT could track each target accurately while KMT fails in tracking target 4 as shown in Fig. 12(c). However, when a target does not have any INs, VRs do not exist and VRT degrades into KMT.

B. Qualitative Comparison Results

From the above analysis, we know that VG could keep targets moving toward their VDs while VR prevents targets from collision when they are getting close to each other. So IKMT which combines the two interaction models could decrease false positives and false negatives during tracking. We compare the performance of our proposed IKMT with KMT based on [17] and 2DLTA which uses LTA [13] model directly as a prediction model of 2-D position for mean-shift tracker [28]. The values provided in [13] for parameters (σ_d and σ_W) in LTA model cannot be used because they are designed for 3-D world coordinate. So we set $\lambda_1 = 2.33$, $\lambda_2 = 2.073$, $\beta = 1.462$, $\alpha = 0.73$, and then try different values for σ_d, σ_W and select the optimal values. The iteration step for gradient descent algorithm is set to be 0.003 and the maximum iteration times is 100.

1) *Experiments on Various Public Datasets:* We select three videos from three different datasets. The first video "fight" is provided in [30]. It contains 299 frames and the size of each frame is 720×576. Four boys run and chase each other in the scene. Two boys wear similar white shirts, which increases some difficulties for the tracking methods. Moreover, occlusion occurs once in a while, which makes multitarget tracking more difficult. The four boys do not have a constant destination and their VDs do not remain the same and change all the time. Thanks to our dynamic VD selection during tracking, we could assign an effective VD for every target. Fig. 13 shows some tracking results using IKMT, KMT, and 2DLTA, respectively, before and after an occlusion. In frame 95, the appearances of target 3 are very similar to its surroundings, so KMT and 2DLTA fail in tracking it [frame 135, Fig. 13(b), (c)]. Benefiting from the proposed interaction models, IKMT achieves much better results as

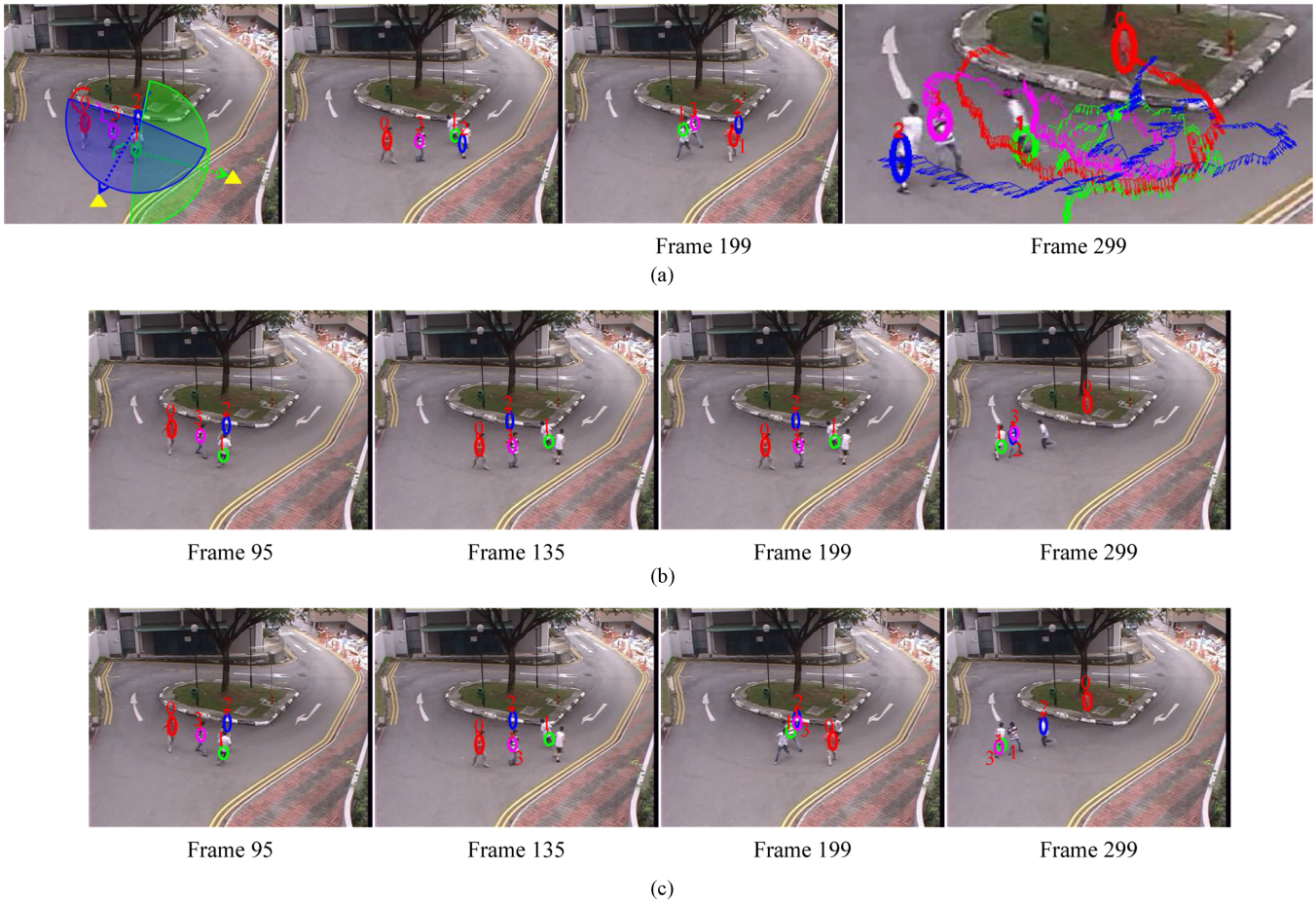


Fig. 13. (a)–(c) Some tracking results using IKMT, KMT, and 2DLTA. Trajectories and velocities obtained by IKMT are shown in frame 299 in (a). Red, green, blue, and pink ellipses correspond to target 0, 1, 2, and 3. (a) Yellow triangles are VDs and target 1 (green ellipse) and target 2 (blue ellipse) are connected with their associated VDs with arrows. Their RAs obtained in our experiment are marked with sectors.

Fig. 14. Tracking results of the third video using IKMT, KMT, and 2DLTA. Yellow triangles are potential VDs and some targets are connected with its associated VD with a dotted arrow in (a). Red, green, blue, pink, and yellow ellipses correspond to targets 0, 1, 2, 3, and 4. (a) Frame 217 IKMT. (b) Frame 217 KKMT. (c) Frame 217 2DLTA.

shown in Fig. 13(a). Since target 2 is in the RA of target 3, he has VR toward target 3. Meanwhile, because target 3's VD (yellow triangle linked to target 3 through a blue arrow) has VG toward him, he would run toward his right and front. This explains why IKMT could track target 3 correctly. It is similar for target 2. In frame 199, two collisions are likely to happen and the targets want to avoid it. IKMT succeeds in tracking them too. In the last frame (frame 299), we provide the trajectories and velocities. We can see that the trajectories of four targets are a little complicated and IKMT achieves very good performance.

The second video is "egtest01" coming from [31]. There are 1821 frames and each frame is 640 by 480. Several similar cars loop around on a road and their appearances vary widely because of sunlight. Several potential VDs represented by yellow triangles are displayed in Fig. 14. They are used dynamically for each target in the experiment. Fig. 14 shows some tracking results of IKMT and KMT. As the road is similar to some cars, without VG, "drift" appears sometimes (e.g., target labeled with red ellipse in frame 217). When two similar cars come close, KMT confuse them while IKMT and 2DLTA overcome the challenge and achieve successful tracking performance as shown in Fig. 14(a) and (c).

The third is "seq_hotel" used as training data in [13]. It contains about 19 349 frames and lasts about 13 min. The

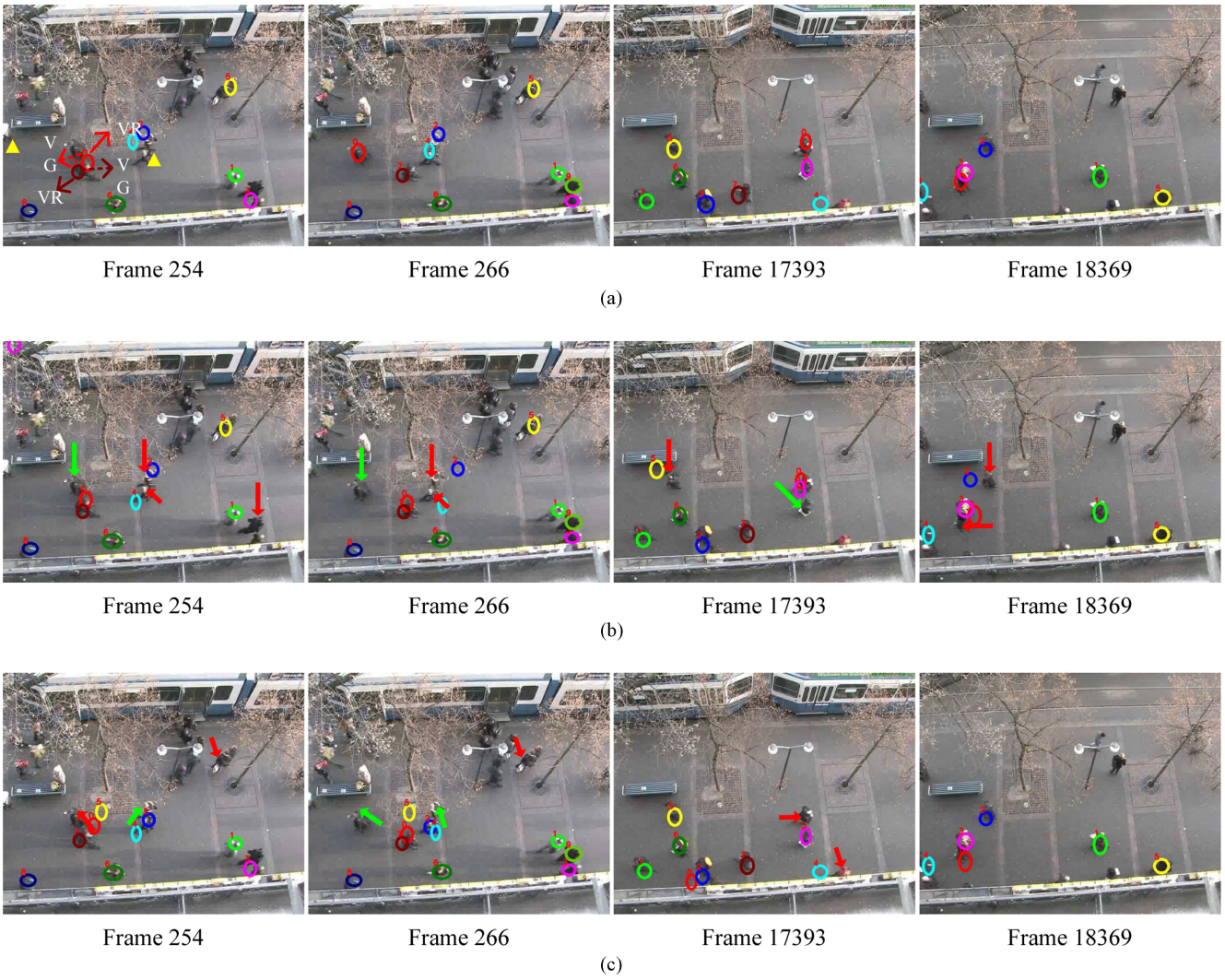


Fig. 15. For the purpose of easy comparison, in (b) and (c), we mark targets that are missed with red arrows and the targets that are mismatched to the other targets with green arrows. (a) Some tracking results of IKMT. In frame 254, the VR and VG imposed on the red target are represented by dotted red arrow and solid red arrow while forces imposed on the dark red target are represented by dark red arrows. (b) Corresponding tracking results of KMT. (c) Corresponding tracking results of 2DLTA.

number of targets varies through the whole image sequence and occlusion happens from time to time. As an example, Fig. 15(a) provides tracking results in several frames with complicated interactions. The first image in Fig. 15(a) shows a very common and simple scenario where two people walk toward each other. When they are close to each other, according to our model, VR and VG affect their motions. Trackers of the two targets would not get too close in case of collision while keep walking toward their VDs. So they can track targets successfully. However, without VG and VR, trackers fail and tracking results begin to drift away as the red arrows shown in the first and second images of Fig. 15(b) and (c). In the right two columns of Fig. 15, some targets' are similar to the background, so KMT and 2DLTA lose the targets. As those targets moved almost in a straight line, their virtual destinations are laid in front of them and the associated VGs attract them to move forward. This is why IKMT could alleviate false positives to some extent.

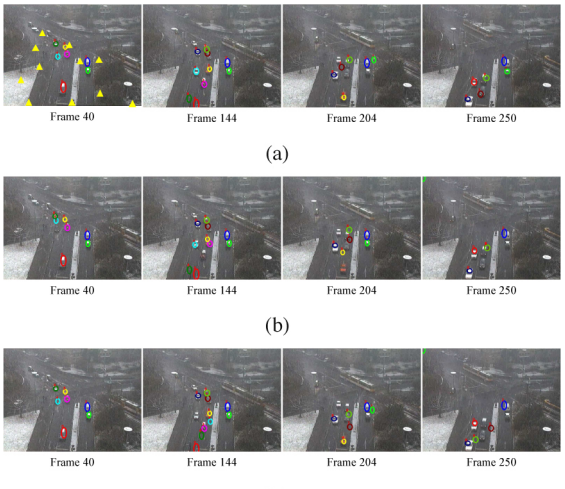


Fig. 16. Some tracking results on "dtneu_schnee" using (a) IKMT, (b) KMT, and (c) 2DLTA.

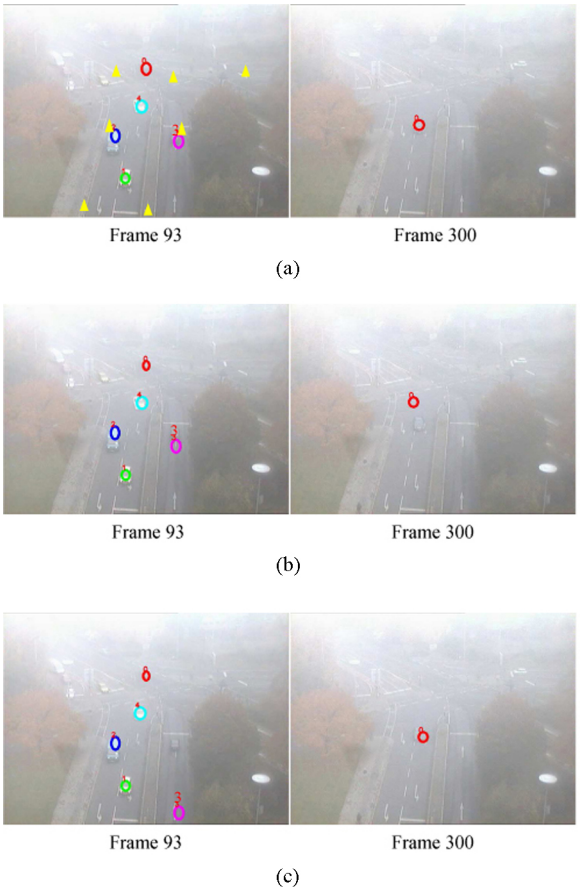


Fig. 17. Some tracking results on "dtneu_nebel" using (a) IKMT, (b) KMT, and (c) 2DLTA.

2) *Experiments on Challenging Traffic Sequences:* We test our tracker on three traffic sequences[2] showing the interaction Karl–Wilhelm–Straβe in Karlsruhe in bad weather. The first one is "dtneu_schnee" which is about Karl–Wilhelm–Straβe when filmed in heavy snow. Some parts of the targets are occluded by the fluttering snowflakes sometime and the top of some cars are covered by snow. This brings great challenges for tracking, as their appearances become less distinguishable. So KMT and 2DLTA fail in tracking some of them as shown in Fig. 16(b) and (c). The second sequence "dtneu_nebel" is recorded by a stationary camera with heavy fog. Every car looks blur especially the dark-colored cars (e.g., target 3 in frame 93 of Fig. 17). Their appearances are so foggy that even people have difficulties to identify them not to mention tracking them with KMT. Fig. 17(b) displays the tracking results using KMT and we can see that the KMT tracker loses the targets and drifts to the road in frame 300. The same thing happens for 2DLTA in frame 93. The third video referred to as "dtneu_winter" shows the interaction after the snowfall. The lanes are covered by thick snow. So when a white car approaches the lane, KMT would drift to the lane because their appearances are very similar, taking target 3 (labeled with pink ellipse) in frame 286 displayed in Fig. 18(b) as an example. Although 2DLTA could track target 3, it loses target 0 in frame 122 and never recovers from the failure.

[2]Available at http://i21www.ira.uka.de/image_sequences.

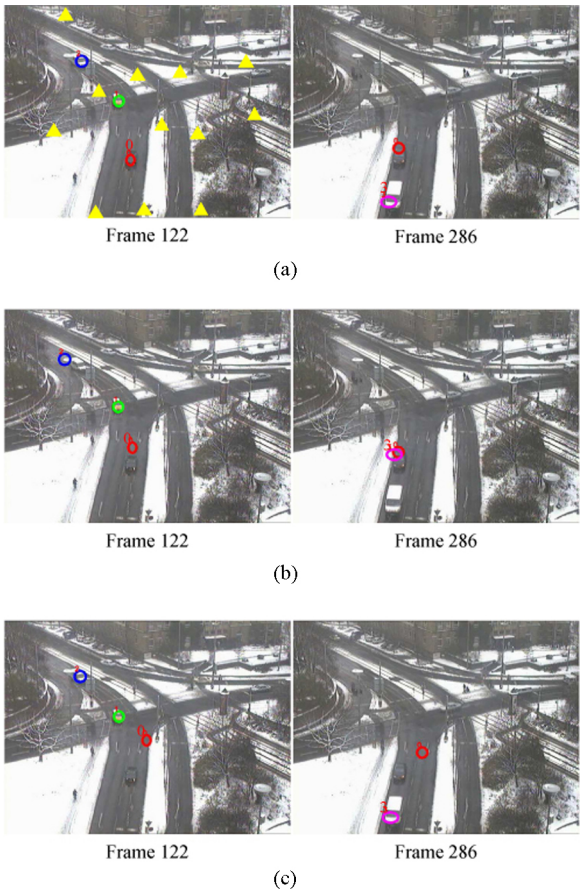


Fig. 18. Some tracking results on "dtneu_winter" using (a) IKMT, (b) KMT, and (c) 2DLTA.

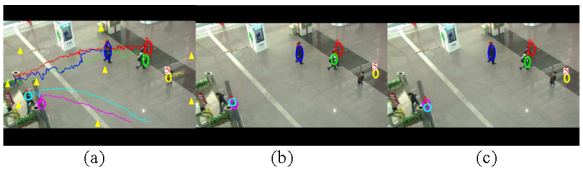


Fig. 19. Tracking results of the "airport" using (a) IKMT, (b) KMT, and (c) 2DLTA. Yellow triangles are potential VDs. Red ellipse: target 0, green: target 1, blue: target 2, pink: target 3, cyan: target 4, yellow: target 5. (a) Frame 1604 IKMT. (b) Frame 1604 KMT. (c) Frame 1604 2DLTA.

In a word, KMT fails in the three challenging sequences. However, IKMT performs well and Figs. 16–18 provide some representative examples. In the first and third sequences, the VG plays a very important role in predicting the states of the tracked targets while in the second sequence both VR and VG guide the tracker to obtain an effective state and accurate tracking results.

3) *Experiments on Videos Recorded by Ourself:* We found that there are little public datasets especially top-view datasets for multitarget tracking, we recorded several videos at airport, parking lot, and residential area road, because those kinds of places play very important roles in our daily lives.[3] We record three different types of videos totally: pedestrian, vehicles, pedestrian and vehicles. Here, we report three representative videos named as "airport," "carpark," and "low-frame rate video," respectively.

[3]Available at http://www.jdl.ac.cn/user/grli.

Fig. 20. Tracking results on "carpark" using (a) IKMT, (b) KMT, and (c) 2DLTA. Trajectories as well as velocities obtained using IKMT are shown in (a). (a) Frame 303 IKMT. (b) Frame 303 KMT. (c) Frame 303 2DLTA.

Fig. 21. Tracking results of "low-frame rate video" using (a) IKMT, (b) KMT, and (c) 2DLTA. Yellow triangles denote potential VDs and the curves are trajectories. Cyan ellipse: target 4, yellow ellipse: target 5. (a) Frame 1888 IKMT. (b) Frame 1888 KMT. (c) Frame 1888 2DLTA.

The first video "airport" is about pedestrians in an airport. It contains 1876 frames with a resolution of 800×608. Although the number of targets varies, all of them move to or away the elevator which is on the left side of every frame. Thus, we could easily discover potential VDs such as the yellow triangles in Fig. 19(a). Sometimes the background is similar to targets (e.g., target 5 in frame 1604) or occlusion happens [e.g., Fig. 12(a)] and KMT as well as 2DLTA fails in tracking them. However, as our interaction models could provide more effective prediction for targets' motion, IKMT performs much better as shown in Fig. 19(a).

The second video "carpark" is recorded at a car park. Each frame is 800×608. Since cars either drive into the park or pass by it, it is very easy to discover several potential VDs. In Fig. 20(a), VDs are displayed as yellow triangles and apparently, all the three targets are moving toward VD A. According to our interactive models, IKMT could derive that cars driving into the park need to change lanes earlier in order to avoid collision with the white car (red ellipse). This helps IKMT tracking those targets successfully as shown in Fig. 20. Since the sunlight makes some cars' appearances look very similar with the road and pillars, KMT and 2DLTA have the occasion to fail, such as target 2 and target 3 in Fig. 20(b) and target 0, 2 in Fig. 20(c). Thanks to VG and VR, IKMT tracks cars successfully.

The third one is a low-frame rate (12.5 f/s) video (denoted as "low-frame rate video") which contains 2037 frames totally with a resolution of 640 by 480. Although the number of targets varies through the whole image sequence, all of them move along the road. Thus, we could easily label their potential VDs such as the yellow triangles in Fig. 21(a). Occlusion also happens sometime in this video. In frame 1888 shown in Fig. 21, both KMT and 2DLTA lose target 4 because it is partially occluded by target 5. However, IKMT still could track it successfully due to exploiting the VRs between these two close targets. Figs. 9(e), 10(e), 11(c), and 12(c) illustrate more tracking results of different trackers.

C. Quantitative Comparisons

In order to make quantitative comparisons among IKMT, VRT, VGT, KMT, and 2DLTA, we evaluate the results of the two trackers with CLEAR MOT metrics[4] [32] and the results are shown in Table III, where MI and FP are short for "Misses" and "False Positives," respectively. The smaller they are, the better tracking result is. For the purpose of easy comparison, we mark the best results in boldface font for each sequence.

With the constraints of VG model, targets move toward their VDs, which could prevent the tracker from drifting to the background. So MI can be reduced. From Table III, we can see that MI of VGT is lower than that of KMT in all the videos except for "egtest01." Meanwhile, our VR model prevents targets from getting too close, so the probability for FP to happen is much lower. The data in Table III shows that, in most videos, FP of VRT is smaller than that of KMT. We also notice that VRT and VGT perform not so good on video "egtest01." MI of VGT and FP of VRT are a little larger than that of KMT and 2DLTA, because this video is recorded with a moving camera. In the end of "egtest01," camera moves faster and the obtained velocities of the targets are very different from their actual velocities in the 2-D frame. So VR and VG are not works in some frames, leading to bad performance.

From Table III, we can see that both VRT and VGT make progress compared to KMT. Combining VRT with VGT, IKMT makes use of both advantages and achieves more satisfactory results than VRT, VGT, KMT, and 2DLTA.

D. Analysis of Unexpected Results

We found that when target's appearance is very similar to background or other targets or occlusion exists, IKMT sometimes fails and Fig. 22 shows some examples. This is because that although the proposed interaction models could provide an effective prediction of the target's future state, the target's appearance model plays the key role when performing measurement update as Step 2.3 described in Table I. It is worth noting that when IKMT drifts to the background, KMT and 2DLTA fail most time. Fig. 23 is an example. However, when KMT and 2DLTA drift to the background, IKMT may not fail and Figs. 14–21 show some examples. Table IV shows our statistical data[5] on tracking results of the nine test videos, and we can see that there are 241 times that IKMT could track the target successfully when KMT fails. On the contrary, there are only 40 times that KMT performs successfully when IKMT loses the target. In other words, in many cases IKMT could insist on tracking targets successfully when KMT or 2DLTA fails, and thus IKMT is more capable of tracking targets for a longer time than KMT and 2DLTA.

To recover or resume the correct tracking from IKMT's failure, first, it needs to judge when the tracker fails. We think there are two categories of methods: one is integrating into a detector and the other is doing inference according to context information. For example, if we want to track people, people detector which is trained before tracking could be used to check whether the obtained tracking result is a person or

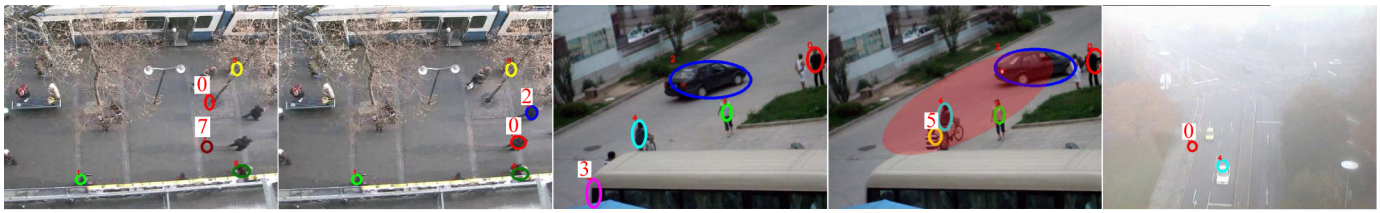
[4]In our experiments, most ID switches happened when we do re-initialization, so the numbers of ID switches generated by IKMT, KMT, and 2DLTA are almost the same. Therefore, we do not use ID switches for evaluation.

[5]If the tracker loses a target in consecutive frames, we consider that it loses that target one time.

TABLE III

CLEARMOT METRICS [32] OF IKMT, VRT, VGT, KMT, AND 2DLTA. THE SMALLER MI AND FP ARE, THE BETTER THE TRACKER IS

	IKMT	VRT	VGT	KMT	2DLTA	IKMT	VRT	VGT	KMT	2DLTA
	"fight"					"egtest01"				
MI	**0**	0.0518	0	0.0518	0.1231	0.0060	**0**	0.2130	0.0096	0.0057
FP	**0**	0.1112	0.1290	0.1961	0.0535	**0**	0.0201	0	0.0174	0
	"seq_hotel"					"dtneu_schnee"				
MI	**0.0647**	0.1131	0.1159	0.1177	0.1642	0.0724	0.1864	**0.0208**	0.1331	0.1392
FP	**0.0229**	0.0248	0.0376	0.0327	0.0348	0.0059	**0.0351**	0.1070	0.1102	0.0383
	"dtneu_nebel"					"dtneu_winter"				
MI	0.0382	0.3468	**0.0281**	0.0988	0.1392	**0.2407**	0.2886	0.2485	0.2886	0.3843
FP	**0**	0	0	0	0	**0**	0	0	0.0355	0
	"airport"					"carpark"				
MI	0.0671	**0.0647**	0.0814	0.0890	0.1359	**0.0101**	0.1227	0.0327	0.1227	0.1948
FP	0.0228	**0.0176**	0.0193	0.0680	0.7470	**0**	0	0	0	0.1115
	"low-frame rate"					Average				
MI	**0.0140**	0.0970	0.1965	0.0234	0.2456	**0.0570**	0.1412	0.1041	0.1039	0.1702
FP	0.0299	**0.0101**	0.0339	0.0979	0.1814	**0.0091**	0.0243	0.0363	0.0620	0.1296



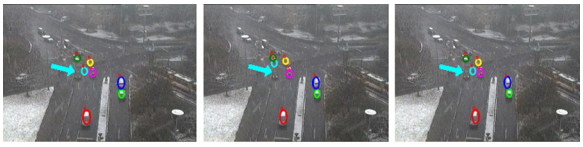
(a) (b) (c) (d) (e)

Fig. 22. Some examples of unexpected results using IKMT. The results on (a) target 0 and target 7, (c) target 3, and (e) target 0 are wrong. (b) Target 0 and 2 are the new re-initialized targets. (d) Target 5 is the re-initialized target.

TABLE IV

STATISTICS OF THE TIMES THAT KMT, 2DLTA, OR IKMT LOSES TARGETS ON THE NINE TEST VIDEOS

Times	KMT Succeed	2DLTA Succeed	Times	KMT Fail	2DLTA Fail
IKMT fail	40	26	IKMT succeed	241	209



(a) (b) (c)

Fig. 23. Example for illustrating that when IKMT fails, KMT and 2DLTA fail too. Target 4 pointed by the cyan bluish arrow is very similar to the road and the three trackers cannot track it successfully. (a) Result of IKMT. (b) Result of KMT. (c) Result of 2DLTA.

not. For example, in Fig. 22(a), by using pedestrian detector, perhaps we could know that the results of target 0 and target 7 are not pedestrians at all. So the tracker must be wrong. It is similar for target 3 in Fig. 22(c). Besides, context information is very useful. Many scene labeling methods [33] have been developed. Before tracking, those methods can be used to label frames and then we could infer that where targets could move (e.g., road) and where targets are unlikely to appear (e.g., sky, wall, and tree). After obtaining tracking results, we could make inference. For example, in Fig. 22(e), with the help of context information, we may know that where the road is. As the result of target 0 shows that the car is not driving on the road, we could derive that the tracker is wrong. Second, when a new target appears, we need to infer whether or not it appeared before. Background subtraction (e.g., [34]–[36]) or object detection methods (e.g., pedestrian detection [37], [38]) are potential methods for automatically discovering and initializing targets for trackers. Then whether the discovered target is tracked before could be inferred based on two kinds of information. One is provided by the target itself, such as appearance, motion. The other is provided by scene context, such as where a new target is likely to appear or disappear. Take target 5 in Fig. 22(d) as an example, if we know new targets are impossible to enter from the area covered by red ellipse, target 5 should not be a new target and we could associate it with a lost target according to their appearance similarity.

V. CONCLUSION

In this paper, we studied a common scenario of multitarget tracking problems, where the tracked targets have both virtual destinations and interactions. By a thorough investigation and deep analysis, two sophisticated interaction models have been proposed to simulate target's motion behavior. Moreover, these two models are further integrated with kernel-based framework and a novel kernel-based multitarget tracking approach is derived. With the constraints of the interactive

models, we first analyzed that IKMT could achieve superior tracking results than KMT in theory. Then further experiments were performed on challenging videos. Both qualitative and quantitative comparisons demonstrate that IKMT could outperform KMT and 2DLTA.

In our paper, we did not consider environmental information such as road, grass, and sky, which will be the topic of our future work. As sometimes objects move in a group and in some cases objects intend to get close, such as two people walk toward each other for handshakes, modeling such kind of behavior of groups is also a promising topic of our future study. Another limitation of our tracker is when background or some other foregrounds look very similar to the targets, IKMT sometimes fail and begin to track the distracter. We think feature selection is a promising solution and in the future we will investigate feature selection methods for multitarget tracking specially.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers, the Associate Editor, and J. Pang for their constructive advice.

REFERENCES

[1] J. Maccormick and A. Blake, "A probabilistic exclusion principle for tracking multiple objects," *Int. J. Comput. Vis.*, vol. 39, no. 1, pp. 57–71, 2000.

[2] W. Qu, D. Schonfeld, and M. Mohamed, "Real-time interactively distributed multiobject tracking using a magnetic-inertia potential model," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2005, pp. 535–540.

[3] M. Han, W. Xu, H. Tao, and Y. Gong, "An algorithm for multiple object trajectory tracking," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2004, pp. 864–871.

[4] K. Okuma, A. Taleghani, N. Freitas, J. Little, and D. Lowe, "A boosted particle filter: Multitarget detection and tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2004, pp. 28–39.

[5] T. Yu and Y. Wu, "Decentralized multiple target tracking using netted collaborative autonomous trackers," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 939–946.

[6] Y. Li, H. Ai, T. Yamashita, S. Lao, and M. Kawade, "Tracking in low frame rate video: A cascade particle filter with discriminative observers of different lifespans," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[7] C. Huang, B. Wu, and R. Nevatia, "Robust object tracking by hierarchical association of detection responses," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 788–801.

[8] A. Perera, C. Srinivas, G. Brooksby, and W. Hu, "Multiobject tracking through simultaneous long occlusions and split-merge conditions," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 666–673.

[9] B. Wu and R. Nevatia, "Detection and tracking of multiple, partially occluded humans by Bayesian combination of Edgelet based part detectors," *Int. J. Comput. Vis.*, vol. 75, no. 2, pp. 247–266, 2007.

[10] Z. Khan, T. Balch, and F. Dellaert, "An MCMC-based particle filter for tracking multiple interaction targets," in *Proc. Eur. Conf. Comput. Vis.*, May 2004, pp. 279–290.

[11] J. Liu and Y. Liu, "Multitarget tracking of time-varying spatial patterns," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1839–1846.

[12] S. Ali and M. Shah, "Floor fields for tracking in high density crowd and scenes," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2008, pp. 1–14.

[13] S. Pellegrini, A. Ess, K. Schindler, and L. Gool, "You'll never walk alone: Modeling social behavior for multitarget tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sep.–Oct. 2009, pp. 261–268.

[14] J. Zhu, Y. Lao, and Y. Zheng, "Object tracking in structured environments for video surveillance applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 2, pp. 223–235, Feb. 2010.

[15] A. Tyagi and J. Davis, "A context-based tracker switching framework," in *Proc. IEEE Workshop Motion Video Comput.*, Jan. 2008, pp. 1–8.

[16] A. Tyagi, "Layered tracker switching for visual surveillance," Ph.D. dissertation, Dept. Comput. Sci. Eng., Ohio State Univ., Columbus, 2008 [Online]. Available: http://etd.ohiolink.edu/send-pdf.cgi/Tyagi%20Ambrish.pdf?osu1218257609

[17] D. Comaniciu, R. Visvanathan, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 564–577, May 2003.

[18] G. Welch and G. Bishop, "An introduction to the Kalman filter," in *Proc. SIGGRAPH*, Aug. 2001, course 8, pp. 19–33.

[19] G. Hager, M. Dewan, and C. Steward, "Multiple kernel tracking with SSD," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun.–Jul. 2004, pp. I-790–I-797.

[20] Z. Fan, M. Yang, and Y. Wu, "Multiple collaborative kernel tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 7, pp. 1268–1273, Jul. 2007.

[21] W. Qu and D. Schonfeld, "Robust control-based object tracking," *IEEE Trans. Image Process.*, vol. 17, no. 9, pp. 1721–1726, Sep. 2008.

[22] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Phys. Rev.*, vol. 51, no. 5, pp. 4282–4286, 1995.

[23] C. Stauffer, "Estimating tracking sources and sinks," in *Proc. IEEE Workshop Event Mining*, Jun. 2003, p. 35.

[24] E. Hall, *The Hidden Dimension*. New York: Doubleday, 1996.

[25] H. Young, R. Freedman, T. Sandin, and A. Ford, *Sears and Zemansky's University Physics*. Reading, MA: Addison Wesley, 1999.

[26] Y. Hu and Y. Guo, *Operational Research Tutorial*. Beijing, China: Tsinghua University Press, 1998.

[27] P. Maybeck, *Stochastic Models, Estimation, and Control*. New York: Academic Press, 1979.

[28] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2000, pp. 142–149.

[29] A. Ramm, *Inverse Problems: Mathematical and Analytical Techniques with Applications to Engineering*. Berlin, Germany: Springer, 2005.

[30] B. Ni, S. Yan, and A. Kassim, "Recognizing human group activities with localized gausalities," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1470–1477.

[31] R. Collins, X. Zhou, and S. The, "An open source tracking testbed and evaluation web site," in *Proc. IEEE Int. Workshop Performance Eval. Tracking Surveillance*, Jan. 2005, pp. 1–8.

[32] K. Bemardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The Clear Mot metric," *EURASIP J. Image Video Process.*, vol. 2008, no. 246309, p. 10, 2008.

[33] P. F. Felzenszwalb and O. Veksler, "Tiered scene labeling with dynamic programming," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3097–3104.

[34] S. C. Cheung and C. Kamath, "Robust background subtraction with foreground validation for urban traffic video," *EURASIP J. Appl. Signal Process.*, vol. 14, pp. 2330–2340, Jan. 2005.

[35] Y. Sheikh, O. Javed, and T. Kanade, "Background subtraction for freely moving cameras," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2009, pp. 1219–1225.

[36] D. Lee, "Effective Gaussian mixture learning for video background subtraction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 827–832, May 2005.

[37] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *Int. J. Comput. Vis.*, vol. 63, no. 2, pp. 153–161, 2005.

[38] G. Gualdi, A. Prati, and R. Cucchiara, "Multistage sampling with boosting cascades for pedestrian detection in image and videos," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 196–209.

Guorong Li received the B.S. degree in technology of computer application from the Renmin University of China, Beijing, China, in 2006. She is currently pursuing the Ph.D. degree with the School of Information Science and Engineering, Chinese Academy of Sciences, Beijing.

She is a Graduate Research Assistant with the Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences. Her current research interests include object tracking, video analysis, pattern recognition, and computer vision.

Wei Qu (S'04–M'06) received the B.S. degree in electrical and computer engineering from the Beijing Institute of Technology, Beijing, China, in 2000, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Illinois, Chicago, in 2005 and 2006, respectively.

He was a Senior Research Scientist with the Motorola Research Laboratory, Schaumburg, IL, from 2004 to 2007, and a Senior Researcher with Siemens Medical Solutions, Inc., Hoffman Estates, IL, from 2007 to 2009. He joined the Chinese Academy of Sciences, Beijing, as an Associate Professor in 2009.

Dr. Qu received the Best Student Paper Award at the IEEE International Conference on Image Processing in 2006. He has served regularly as a reviewer for different journals and conferences, such as the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE TRANSACTIONS ON IMAGE PROCESSING, and others.

Qingming Huang (M'04–SM'08) received the Ph.D. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 1994.

He was a Post-Doctoral Fellow with the National University of Singapore, Singapore, from 1995 to 1996, and was with the Institute for Infocomm Research, Singapore, as a Member of Research Staff from 1996 to 2002. He joined the Chinese Academy of Sciences, Beijing, China, under the Science100 Talent Plan in 2003, and is currently a Professor with the Graduate University, Chinese Academy of Sciences. His current research interests include image and video analysis, video coding, pattern recognition, and computer vision.