Multiple Object Tracking: A Review

WENHAN LUO, XIAOWEI ZHAO and TAE-KYUN KIM, Department of Electrical and Electronic Engineering, Imperial College London

Multiple Object Tracking (MOT) is an important computer vision task which has gained increasing attention due to its academic and commercial potential. Although different kinds of approaches have been proposed to tackle this problem, it still has many issues unsolved. For example, factors such as continuous appearance changes and severe occlusions result in difficulties for the task. In order to help the readers understand and learn this topic, we contribute a comprehensive and systematic review. We review the recent advances in various aspects about this topic and propose some interesting directions for future research.

To our best knowledge, there has not been any review about this topic in the community. The main contribution of this review is threefold: 1) All key aspects in the multiple object tracking system, including what scenarios the researchers are working on, how their work can be categorized, what needs to be considered when developing a MOT system and how to evaluate a MOT system, are discussed in a clear structure. This review work could not only provide researchers, especially new comers to the topic of MOT, a general understanding of the state-of-the-arts, but also help them to comprehend the aspects of a MOT system and the inter-connected aspects. 2) Instead of listing and summarizing individual publications, we categorize the approaches in the key aspects involved in a MOT system. In each aspect, the methods are divided into different groups and each group is discussed in details for the principles, advances and drawbacks. 3) We provide some potential directions with insights for MOT, which are still open issues and need more research efforts. This would be helpful for researchers to identify further interesting problems.

Categories and Subject Descriptors: I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis—Tracking

General Terms: Algorithms, Performance

Additional Key Words and Phrases: Multi-object tracking, data association

1. INTRODUCTION

Multiple Object Tracking (MOT), or Multiple Target Tracking (MTT), plays an important role in computer vision. The task of MOT is largely partitioned to locating multiple objects, maintaining their identities and yielding their individual trajectories given an input video. The interesting objects to track in MOT should be homogeneous in the video. For example, the multiple objects can be pedestrians on the street, sport players in the court, cars, or a flock of animals (birds, fishes, etc.).

As a mid-level task in computer vision, multiple object tracking grounds high-level tasks such as action recognition, behavior analysis, etc. It has numerous applications. Some of them are presented in the following.

— Visual Surveillance. The massive amount of videos, especially surveillance videos, requires automatic analysis to detect abnormal behavior, which is based on analysis of the objects' actions, trajectories, etc. To obtain such information, we need to locate and track them, which is exactly the task of multiple object tracking.

DOI:http://dx.doi.org/10.1145/0000000.0000000

— Human Computer Interface (HCI). Visual information, such as expression, gesture, can be employed to achieve advanced HCI. Extraction of visual information requires visual tracking as the basis. When multiple objects appear in the scene, we need to consider interaction among the multiple objects. In this case, MOT plays a crucial rule to make the HCI more natural and more intelligent.
— Virtual Augment Reality (VAR). MOT also has an application for this. For example, MOT can supply the users with better experience in video conferences.
— Medical Image Processing. Some medical tasks require laborious manual labeling, for instance, labeling multiple cells in images. In this case, MOT can help save a large amount of labeling cost.

The various applications above have sparkled a large research interest in this topic. However, compared with Single Object Tracking (SOT) which focuses on designing sophisticated appearance models or motion models to deal with challenging factors such as scale changes, out-of-plane rotations and illumination variations, multiple object tracking additionally requires maintaining the identities among multiple objects. Besides the common challenges in both SOT and MOT, the further key issues making MOT challenging include: 1) frequent occlusion due to the crowd, 2) tracks initialization and termination, 3) the size of objects in MOT which is often very small [Betke et al. 2007], 4) similar appearance among objects, and 5) interaction among multiple objects. In order to deal with the MOT problem, a wide range of solutions have been proposed in recent years. These solutions, although they have been partially successful on the challenges aforementioned, focus on different aspects of a MOT system, making it difficult for researchers especially new comers to MOT to gain a comprehensive understanding of this problem. Thus, we categorize and present detailed discussions of the various aspects of a MOT system jointly in this review work.

1.1. Differences from Other Related Reviews

To the best of our knowledge, there has not been any comprehensive literature review on the topic of multiple object tracking. However, there have been some other reviews related to multiple object tracking. We list them in Table I. We group these surveys into two sets and highlight the differences from ours as the following. The first set [Zhan et al. 2008; Hu et al. 2004; Kim et al. 2010; Candamo et al. 2010] involves crowd, i.e., multiple objects. Their focuses are different. This intends to review all the related aspects in developing a multi-object tracking system. In comparison, tracking is only discussed as one part in [Zhan et al. 2008; Hu et al. 2004; Kim et al. 2010; Candamo et al. 2010; Wang 2013]. More specifically, Zhan et al. [2008] focuses on crowd modeling, thus object tracking is only the step to obtain crowd information feature for crowd modeling. The surveys of [Hu et al. 2004; Kim et al. 2010] discuss papers about building a surveillance system for high-level vision tasks, such as behavior understanding, so tracking is an intermediate step. Candamo et al. [2010] review publications about behavior recognition in a special scenario, i.e., transit scenes. In that review, object tracking is discussed as only a core technology as well as motion detection and object classification. Multiple object tracking is also discussed as one module for video surveillance under multiple cameras [Wang 2013]. The second set [Forsyth et al. 2006; Cannons 2008; Yilmaz et al. 2006; Li et al. 2013] is dedicated to general tracking techniques [Forsyth et al. 2006; Cannons 2008; Yilmaz et al. 2006] or some special issues such as appearance models in visual tracking [Li et al. 2013]. Their scope is wider than this review while our review is more comprehensive and detailed in multiple object tracking.

Table I. Summary of other literature reviews

Reference	Topic	Year
[Zhan et al. 2008]	Crowd Analysis	2008
[Hu et al. 2004]	Object Motion and Behaviors	2004
[Kim et al. 2010]	Intelligent Visual Surveillance	2010
[Candamo et al. 2010]	Behavior Recognition in Transit Scenes	2010
[Wang 2013]	Multi-camera video surveillance	2013
[Forsyth et al. 2006]	Human Motion Analysis	2006
[Cannons 2008]	Visual Tracking	2008
[Yilmaz et al. 2006]	Object Visual Tracking	2006
[Li et al. 2013]	Appearance Models in Object Tracking	2013

1.2. Contributions

The main contribution of this review is threefold:

— All the key aspects in the multiple object tracking system, including what scenarios researchers are working on, how their work can be categorized, what needs to be considered when developing a MOT system and how to evaluate a MOT system, are discussed in a systematic manner from the perspective of understanding a topic. We believe in that this could not only provide researchers, especially new comers to the topic of MOT, a general understanding of the state-of-the-arts, but also help them to comprehend the aspects of a MOT system and the inter-connected aspects.
— Instead of enumerating previous works, we categorize the approaches in all aspects involved in a MOT system. In each aspect, the methods are divided into different groups and each group is discussed in details for the principles, advances and drawbacks.
— We provide some potential directions and respective discussions about MOT, which are open research issues.

1.3. Organization of This Review

The goal of this review is to provide a handbook for all the major aspects which the readers need to consider to build a system for tracking multiple objects, step by step. These aspects include what is the current state of research about MOT, all the detailed issues requiring consideration in building the system, and how to evaluate a MOT system. According to this, the organization of this review is as shown in Figure 1. Sec. 2 illustrates the preliminaries (Sec. 2.1), application scenarios (Sec. 2.2), and categorization of MOT (Sec. 2.3). Sec. 3 contributes to some issues involved in multi-object tracking, specifically, tracking framework (Sec. 3.1), appearance model (Sec. 3.2), motion model (Sec. 3.3), interaction model (Sec. 3.4), exclusion model (Sec. 3.5), and occlusion handling (Sec. 3.6), respectively. Furthermore, some special cases of multi-object tracking are described in Sec. 4. The issues concerning evaluations of a MOT system, including metrics (Sec. 5.1), public data sets (Sec. 5.2) and public algorithms (Sec. 5.3) are discussed in Sec. 5. To this end, conclusions are drawn and some interesting directions are provided in Sec. 6.

2. MOT PROBLEM

2.1. MOT Preliminaries

Before beginning the review on the topic of MOT, some preliminaries are given below, to make the review easy to follow.

— *Detection response.* Detection responses are also known as detection observations or detection hypotheses, which are the outputs of an object detector trained for a specific kind of objects, e.g. human, vehicle, face or animal. They are configurations of objects including positions, sizes, etc, in an image sequence.

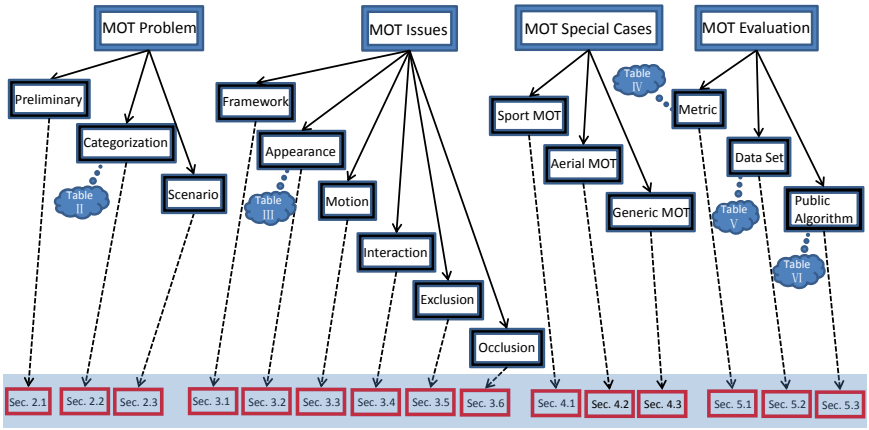


Fig. 1. Organization of this review.

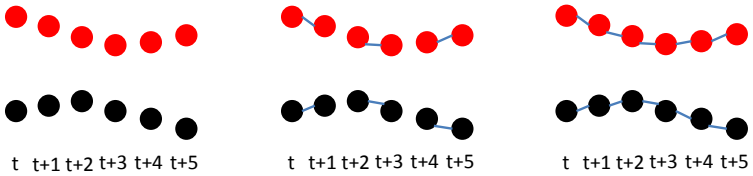
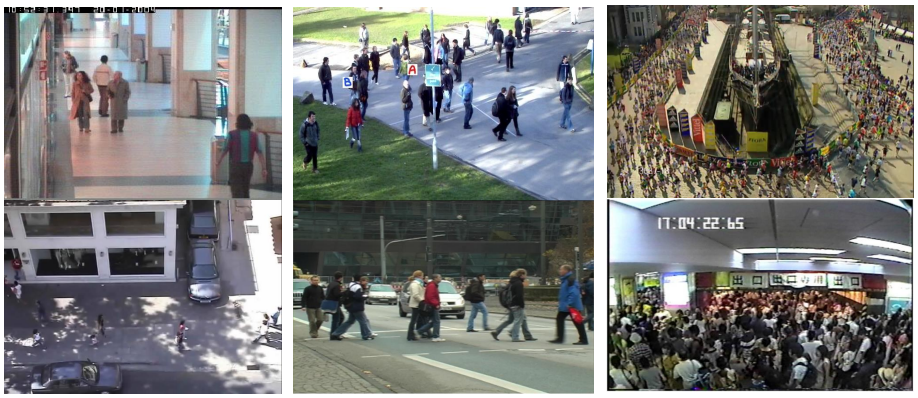


Fig. 2. Detection responses (left), tracklets (center), and trajectories (right) are shown in continuous 6 frames. Different colors encode different targets. Best viewed in color.

— *Trajectory.* Trajectory is the final output of a MOT system. One trajectory corresponds to one target, thus a trajectory is identical. At the same time, one trajectory is composed of multiple continuous object responses of one target in an image sequence, each representing the location, size and some other information in one frame.

— *Tracklet.* Trackelet is an intermediate level of output between detection responses and trajectories. It is composed of several detection responses which are believed to be from the same target. As a fact, a detection response can be viewed as a tracklet composed of only one detection response. The traclet is usually obtained by linking *confident* detection responses, thus it is shorter than trajectory regarding the time span. In some approaches, the final trajectories are obtained by progressively linking detection responses into longer and longer tracklets and eventually forming the trajectories. Figure 2 shows these three concepts.

— *Detection vs Tracking.* Roughly speaking, detection means localizing objects in images. It does not involve any temporal information. Tracking is different, as its task is to localize the same object in continuous frames. Thus tracking always involves in a video or image sequence. The classical visual tracking problem does not necessarily involve detection, as it only requires an initialization in the first frame and then localizes the object in subsequent frames. However, for multiple object tracking, the number of objects is varying since objects may appear or disappear. It is impractical to provide the initialization for each object appearing in the image sequence. In order to automatically initialize an object once it appears, we should resort to an object detector.



(a) (b) (c)

Fig. 3. Examples of (a) under-crowded scenery, (b) semi-crowded scenery and (c) over-crowded scenery. From left to right, top to bottom, images are from CAVIAR data set, PETS2009 data set, UCF Crowds data set, Zara data set, TUD data set and [Lerner et al. 2007].

2.2. MOT Scenarios

We largely categorize the scenarios where the MOT problem arises into three types according to the density: under-crowded, semi-crowded, and over-crowded scenarios. Figure 3 shows some examples of them. A research trend is shifted from the under-crowded scenery to the semi-crowded scenery and then to the over-crowded scenery. One insight behind this is that the component techniques have become maturer to tackle the more challenging crowded scenarios.

— *Under-crowded*. This is the focus of the earlier work of MOT (especially for the CFT approaches (Sec. 2.3.2)). MOT approaches applied in this scenery aim to track multiple objects with minimum tracking pixel error, sometimes to reason the occlusion [Hu et al. 2012] among objects. Regarding the methodology, delicate appearance models, or occlusion reasoning mechanisms are designed to achieve the goal.
— *Semi-crowded*. This is currently the most popular scenery often tackled. An example is the surveillance scenery. As there are always multiple people or pedestrians in this scenery, it is usually dealt with by utilizing the tracking-by-detection strategy. More specifically, most of the researchers employ an off-line trained detector to conduct detection in every frame at first, and try to link all the detection responses into short tracklets and then longer and complete trajectories. In this way, the MOT problem is transformed into a data association problem, which is then solved by some optimization techniques.
— *Over-crowded*. This is a relatively new focus of MOT. The over-crowded scenery typically exhibits frequent partial or complete occlusion, which would confuse the tracker or the data association procedure. The biggest challenge for MOT in the over-crowded scenery is that one object is often not clearly distinguishable from other objects even by the human eye. In this case, the tracking problem cannot be decomposed into two stages of detection and data association, which is the solution in the semi-crowded scenery. Alternatively, researchers treat multiple objects as a whole, mine the properties such as motion patterns [Kratz and Nishino 2010; 2012] or motion structures [Zhao et al. 2012] of the crowd, and utilize this kind of information to alleviate the confusion. Also in [Ali and Shah 2008], the scene structure (exit, barrier, etc.) and the behavior of the crowd as a whole are taken into consideration for MOT in the high density scenery.

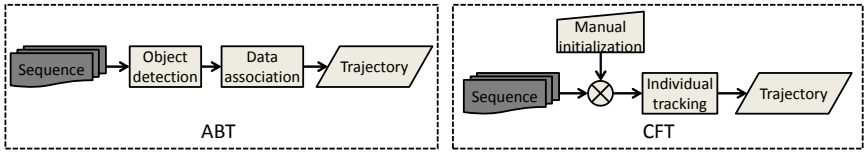


Fig. 4. Procedure flow of ABT (left) and CFT (right).

2.3. MOT Categorization

Following [Yang and Nevatia 2012c], we adopt the view whether the number of objects to be tracked is fixed or not for categorization of MOT approaches. Based on this criterion, MOT approaches could be grouped into two sets: Association Based Tracking (ABT) approaches and Category Free Tracking (CFT) approaches. The number of objects is varying over time in ABT while constant for each frame in CFT. Note that there is also another view to categorize MOT approaches into sequential ones and batch ones [Shi et al. 2013; Luo et al. 2014]. The latter view of categorization is closely related to the former view (see the following discussions), here we discuss the categorisation into ABT and CFT as follows.

2.3.1. ABT. The ABT approaches usually first localize objects in each frame and then link these object hypotheses into trajectories without any initialization labeling. Figure 4 (left) shows the flow of ABT approaches. Given a sequence, object detection or background modeling [Bose et al. 2007; Song et al. 2010] is applied in each frame to obtain detection hypotheses, then data association is conducted to link detection hypotheses into trajectories. There are three issues worthy noting. First, usually the object detection procedure is not the focus of ABT methods. The majority of ABT approaches build upon a pre-trained object detector which produces object hypotheses as observations for data association. Second, as mentioned above, since the object detector which the ABT approaches rely on is pre-trained, ABT approaches focus on specific kinds of targets, such as pedestrians, vehicles or faces. The reason behind is that detection of these objects has gained great progress in the recent years [Dalal and Triggs 2005; Felzenszwalb et al. 2010; Sun et al. 2006]. Third, the performance of ABT approaches depends on the performance of the employed object detector to a certain extent. This could also be a drawback of this kind of approaches.

2.3.2. CFT. As shown in Figure 4 (right), the CFT approaches [Hu et al. 2012; Zhang and van der Maaten 2013b; 2013a; Yang et al. 2007] require the initialization of a fixed number of objects in the first frame (in the form of bounding boxes or other shape configurations), then localize these fixed number of objects in the subsequent frames. They do not rely on object detector to provide object hypotheses. From this point, the CFT methods can be viewed as the extension of single object tracking to track multiple objects, i.e., multiple parallel single object trackers. The key difference between CFT and multiple parallel single object trackers is that CFT methods investigate the relationship among the multiple objects to improve the overall performance. Due to the typical way of doing tracking, CFT methods cannot always obtain a global solution and they quite focus on individual objects.

2.3.3. Discussions. ABT solutions are popular for the fact that they can automatically discover new objects and terminate tracking of existing objects when they dissapear. At the same time, since all the detection hypotheses are given in advance for data association in ABT approaches, the final solution could be globally optimal. CFT requires manual initialization of each object to be tracked, thus it cannot deal with the case when new objects appear. However, it is model-free, i.e., free of pre-trained object de-

Table II. Comparison between ABT approaches and CFT approaches. Part of this table is from [Yang and Nevatia 2012c]

Item	ABT	CFT
Initialization	automatic, imperfect	manual, perfect
# of objects	varying	fixed
Track solution	global	individual
Applications	specific type of objects	any type of objects
Advantages	ability to handle varying number of objects	free of object detector
Drawbacks	performance depends on object detection	requires manual initialization

tectors. So it can deal with sequences of any type of objects. We summarise the major differences between ABT and CFT in Table II.

As aforementioned, the insights behind the two different categorizations are related. The difference between CFT and ABT is whether an object detector is adopted (ABT) or not (CFT). The key to differentiate sequential and batch methods is in the way that they process observations. The readers may question whether CFT is the sequential method because CFT always processes observations sequentially. That is true because the CFT is free of object detector. It cannot attain future observations, and it can only follow the sequential way. Another vagueness may rise between ABT and the batch approaches, as ABT usually adopts the batch way in associating tracklets or detection responses. Note that there are also sequential ABT approaches which conduct association between previously obtained trajectories and new detection responses.

3. MOT ISSUES

There are a few important issues involved in MOT: tracking framework, appearance model, motion model, interaction model, exclusion model, occlusion handling, which are presented in the following.

3.1. Tracking Framework

Tracking framework is the strategy to address the MOT problem throughout the algorithm. Due to the large diversity of all the frameworks, we only list some popular and well-studied frameworks as follows.

3.1.1. Probabilistic framework

— *Bayesian filter*. Bayesian filter is also known as recursive Bayesian estimation. It has wide applications in state-evolving tasks. In terms of MOT, the state to be estimated is usually the trajectory configuration. Observation model is usually on top of visual appearance or motion. This model as well as it variants have been commonly employed to deal with multi-object tracking [Kratz and Nishino 2010; Fortmann et al. 1983; Giebel et al. 2004], such as Kalman filter [Rodriguez et al. 2011; Reid 1979], Extended Kalman filter [Mitzel and Leibe 2011], Particle filter [Jin and Mokhtarian 2007; Yang et al. 2005; Hess and Fern 2009; Han et al. 2007; Hu et al. 2012; Liu et al. 2012; Breitenstein et al. 2009; Yang et al. 2009]. Typically, the probabilistic tracking framework firstly acquires probability formula, then the popular strategy of Maximum A Posteriori (MAP) [Liu et al. 2012; Breitenstein et al. 2009; Yang et al. 2009; Mitzel and Leibe 2011; Rodriguez et al. 2011; Kratz and Nishino 2010; Reid 1979] is adopted to derive a state with the maximum probability.

3.1.2. Graph based framework

— *Network flows*. Network flow is known as the transportation network. It is a directed graph where each edge has capacity. In the MOT problem, nodes in the graph for network flow are usually low-level observations, which could be the detection responses or the short tracklets. Usually the flow is modeled as the indicator to link two nodes

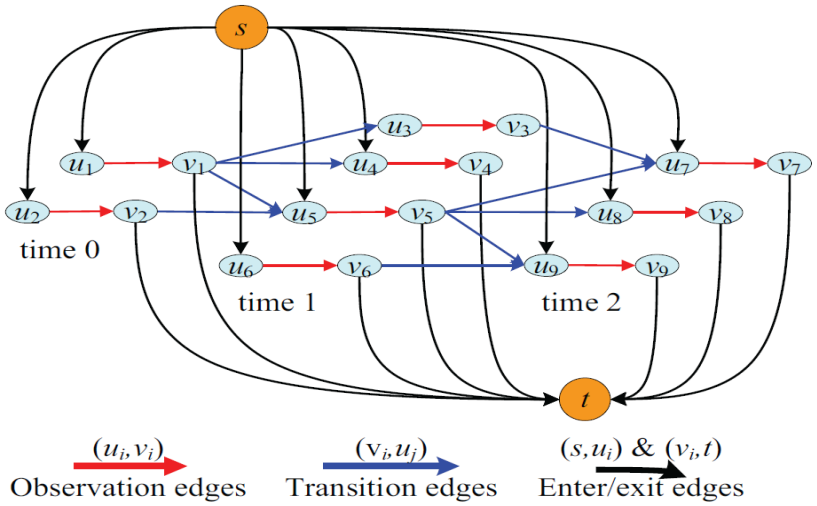


Fig. 5. An image from [Zhang et al. 2008] shows an example of the cost-flow network with 3 timesteps and 9 observations.

(flow is 1) or not (flow is 0). To meet the flow balance requirement, a source node corresponding to the start of a trajectory and a sink node corresponding to the end of a trajectory are added to the original graph (see Figure 5). One trajectory corresponds to one flow path in the graph. The flow transited from the source node to the sink node equals to the number of trajectories or the objects in the video, and the cost to transit the flow from the source node to the sink node is the neg-likelihood of all the association hypotheses. Some examples [Zhang et al. 2008; Choi and Savarese 2012; Wu et al. 2012; Butt and Collins 2013; Pirsiavash et al. 2011] adopt this for MOT problem. To obtain the association result, the min-cost strategy [Wu et al. 2012; Zhang et al. 2008; Butt and Collins 2013], the K-shortest path [Berclaz et al. 2011; Choi and Savarese 2012] and the set cover [Wu et al. 2011] are employed to solve data association problem.

— *Conditional Random Field.* The Conditional Random Field model is adopted to handle the multiple object tracking problem [Yang and Nevatia 2012b; Yang et al. 2011; Milan et al. 2013]. Defining a graph $G = (V, E)$ where V is the set of vertexes and E is the set of edges between vertexes, low level tracklets are given as input to the graph. Each node in the graph is defined as a pair of tracklets, and a label is predicted to indicate whether this pair of tracklets can be linked (label is 1) or not (label is 0). These labels compose the label map which corresponds to the optimal association of the tracklets for the MOT problem.

— *MWIS.* The maximum-weight independent set (MWIS) is the heaviest subset of non-adjacent nodes of an attributed graph. Concerning the MOT problem, the nodes in the attribute graph represent pair of tracklets in continuous frames, weight of the nodes represents the affinity of the pair of tracklets, and the edge is connect if two tracklets share the same detection. Given this graph, the data association problem is modeled as the MWIS problem [Shafique et al. 2008; Brendel et al. 2011].

3.1.3. Others. Some other strategies like the Greedy Bipartite algorithm [Shu et al. 2012], Quadratic Boolean Programming [Leibe et al. 2007], Linear Program (LP) [Jiang et al. 2007], Hungarian algorithm [Qin and Shelton 2012; Reilly et al.

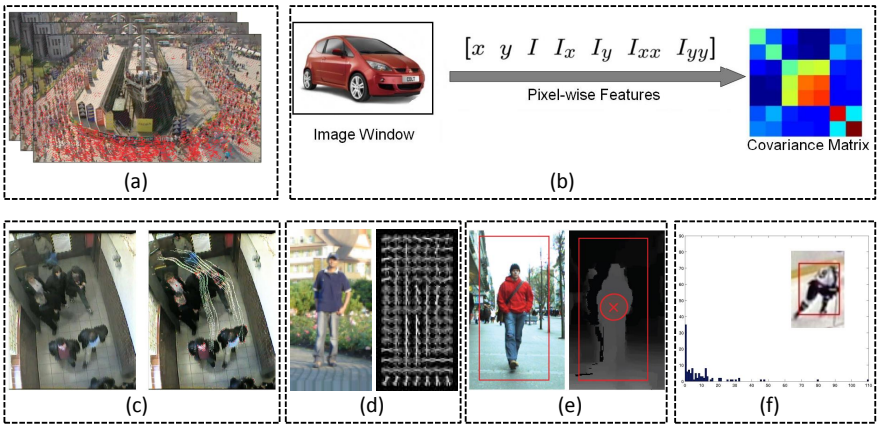


Fig. 6. Some examples to illustrate the features. (a) Image from [Ali and Shah 2008] to show optical flow, (b) image to show covariance matrix, (c) image from [Brostow and Cipolla 2006] to show point feature, (d) image from [Dalal and Triggs 2005] to show gradient based features, (e) image from [Mitzel et al. 2010] to show depth and (f) image from [Okuma et al. 2004] to show color feature. Best viewed in color.

2010; Perera et al. 2006; Xing et al. 2009; Huang et al. 2008] are also employed to solve the MOT problem.

3.2. Appearance Model

Appearance is an important cue for affinity computation in MOT. However, it is worthy noting that, different from single object tracking problem which mainly focuses on constructing a sophisticated appearance model to discriminate the object from the background, multiple object tracking does not mainly focus on the appearance model, i.e., appearance cue is important but not the only cue to depend on. This is partly because that there are many similar objects in MOT which cannot be discriminated by relying on only the appearance information. Thus solutions to MOT do not pay much effort on designing a delicate appearance model.

Technically, appearance model includes two components, i.e. visual representation and statistical measuring. Visual representation is closely related to features, but it is more than features. It is how to precisely describe the visual characteristics of the object based on features, and in general it can be grouped into two sets according to the cues it relies on, visual representation based on single cue and that based on multiple cues. Regarding statistical measuring, it is the computation of similarity or dissimilarity between different observations after the step of visual representation. Eq. 1 gives an illustration of appearance modeling, where O_i and O_j are visual representation of different observations based on single cue or multiple cues, and $H(\bullet, \bullet)$ is the function/metric to measure the similarity S_{ij} between O_i and O_j (sometimes this function needs to transform distance to similarity if necessary). In the following, we firstly discuss the features/cues employed in MOT, and then describe appearance models based on single cue and multiple cues respectively.

$$S_{i,j} = H(O_i, O_j) \tag{1}$$

3.2.1. Feature. Feature is indispensable for MOT (especially for appearance modeling). As shown in Figure 6, different kinds of features have been employed in MOT. We categorize the features into the following sub-sets.

— *Point based features.* Point based features are successful in single object tracking [Shi and Tomasi 1994]. For MOT, point features can also be helpful. For instance, Sugimura et al. [2009] employ the KLT tracker to track feature points and generate a set of trajectories. In [Zhao et al. 2012], KLT keypoint tracker is also employed to obtain short tracklets. Choi and Savarese [2010] utilize the KLT features [Tomasi and Kanade 1991] as additional features to estimate the camera's motion, which greatly improve the tracking performance. Similarly, KLT tracking is utilized in [Benfold and Reid 2011] to estimate motion. Local feature points [Lowe 2004] are adopted along with the bag-of-words model in [Yang et al. 2009] to capture the texture characteristics of a region. Point features are also employed in [Brostow and Cipolla 2006] for motion clustering.
— *Color/intensity based features.* This is the most popularly utilized feature for MOT due to its simplicity. Usually the color or intensity features along with a measurement are employed to calculate the affinity between two counterparts (detection hypotheses, tracklets, short trajectories). In [Yamaguchi et al. 2011], the simple pixel intensity template is employed to compute the appearance affinity. Color histogram is used in [Sugimura et al. 2009; Song et al. 2010; Mitzel et al. 2010; Izadinia et al. 2012; Okuma et al. 2004; Mitzel and Leibe 2011].
— *Optical flow.* The optical flow feature can be employed to conduct short-term visual tracking. Thus many solutions to the MOT problem utilize the optical flow to link detection responses in continuous frames into short tracklets for further data association processing [Rodriguez et al. 2009] or directly use it for data association [Izadinia et al. 2012]. Besides this, optical flow is also employed to complement HOG for observation model [Andriyenko and Schindler 2011]. Optical flow is popular in extremely crowded scenarios to discover crowd motion pattens [Ali and Shah 2008; Rodriguez et al. 2011].
— *Gradient/pixel-comparison based features.* There are some features based on gradient or pixel-comparison. Mitzel et al. [2010] utilize a variation of the level-set formula, which integrates three terms penalizing the deviation from the foreground and background model, the embedding function from a signed distance function and the length of the contour to track objects in continuous frames. For the success of HOG [Dalal and Triggs 2005] in human detection, it plays a vital role in the multiple pedestrian tracking problem. For instance, HOG is employed [Izadinia et al. 2012; Kuo et al. 2010; Breitenstein et al. 2009; Choi and Savarese 2012; Yu et al. 2008] to detect objects and/or compute similarity between pedestrian detections for data association.
— *Region covariance matrix features.* Region covariance matrix [Porikli et al. 2006; Tuzel et al. 2006] features are robust to issues such as illumination changes, scale variations, etc. Therefore, it is also helpful for the MOT problem. The region covariance matrix dissimilarities is employed to compare appearance for data association [Henriques et al. 2011]. Also in [Kuo et al. 2010] covariance matrices along with the other features constitute the feature pool. Hu et al. [2012] utilize the covariance matrix to represent the object for both single and multiple object tracking.
— *Depth.* As depth information can supply with additional cue for various computer vision tasks, it is popularly employed. With regard to MOT, Mitzel et al. [2010] utilize the depth information to correct the bounding box of the detection response and re-initialize the bounding box for the level-set tracking in their work. Depth information is integrated into the framework [Ess et al. 2009; Ess et al. 2007] to augment the detection hypotheses with a depth flag 0 or 1, which further refines the detection responses. Similarly, Ess et al. [2008] employ depth information to obtain more accurate object detections in a mobile vision system and then use the result detection for multiple object tracking. In [Giebel et al. 2004] the stereo depth is taken into ac-

count to estimate the weight of a particle in the proposed Bayesian framework for multiple 3D object tracking. Gavrila and Munder [2007] integrate depth to generate detections and consequently verify them for multiple object tracking from a moving car.

— *Others.* Some other features, which are not so popular but still helpful, are also utilized to conduct multiple object tracking. For instance, the gait features in the frequency domain, which are unique for every person, are employed in [Sugimura et al. 2009] to maximize the discrimination between the tracked individuals. Given a trajectory, a line fitting via linear regression is conducted to extract the periodic component of the trajectory. Then the Fast Fourier Transform (FFT) is applied to the residual periodic signal to obtain the amplitude spectra and phase of the trajectory, which will be utilized to compute the dissimilarity between a trajectory and other trajectories. The Probabilistic Occupancy Map (POM) [Fleuret et al. 2008; Berclaz et al. 2011] is employed to estimate how probable an object would occur in a specific grid under the multi-camera settings, which creates detection hypotheses on top of background modeling as the input for MOT.

Generally speaking, most of the features are simple and efficient, such as the point based features, color/intensity based features, gradient based features. At the same time, they also have shortcomings. For instance, color histogram has well studied similarity measures, but it ignores the spatial layout of the region. Point based features are efficient, but sensitive to issues like occlusion and out-of-plane rotation. Gradient based features like HOG can describe the shape information of object and robust to issues such as illumination changes, but it cannot handle occlusion and deformation well. Region covariance matrix features are more robust as they take more information in account, but this benefit is obtained at the cost of more computation. Depth based features can help to make the computation of affinity more accurate, but they require multiple views of the same scenery and additional algorithm [Felzenszwalb and Huttenlocher 2006] to obtain depth.

3.2.2. Single cue based appearance model. As mentioned before, the appearance model is usually not the most important part of the system. In most cases, it is kind of simplicity and efficiency, thus a single cue is a popular option for most of the MOT approaches. In the following we present the appearance model employing a single cue in five aspects.

— *Raw pixel template representation.* Pixel is the most granular element of images and videos, thus pixel is the foundation of computer vision problems. Beside its importance, it is also popular for its simplicity. The raw pixel template representation is the raw pixel intensity or color of a region to describe. It can encode the spatial information since the comparison is element-wise when matching two templates. Yamaguchi et al. [2011] employ the Normalized Cross Correlation (NCC) to evaluate the predicted position of object as in Eq. 2, where $\mathbf{p}$ is the position of the predicted bounding box, $NCC(\bullet)$ is the normalized cross correlation of pixels corresponding to $\mathbf{p}$. This appearance model is very simple, but helpful.

$$S_{app}(\mathbf{p}) \propto \exp(-\frac{(1-NCC(\mathbf{p}))^2}{2\sigma^2}). \tag{2}$$

In [Ali and Shah 2008], the appearance affinity is calculated as the NCC between the target template and the candidate bounding box. Note that the target template is progressively updated at each time instant. Wu et al. [2012] build a network-flow approach to handle multiple target tracking. When they compute the transitional cost on the arcs of the network as flows, the normalized correlation between the upper one-fourth bounding boxes of the corresponding two detection observations is used.

In [Pellegrini et al. 2009], a simple patch-based tracker is implemented to calculate the data likelihood with regard to appearance. The similarity is computed between the initial bounding box and the candidate bounding box as the squared exponential $P_{data}(\mathbf{p}) \propto \exp(-(NCC(\mathbf{p}, \mathbf{p}^0) - 1)^2)$, where $\mathbf{p}$ and $\mathbf{p}^0$ represent the candidate bounding box and the initial bounding box. Despite of efficiency, this kind of representation has the drawback that it easily suffers from the change of illumination, occlusion or some other issues.

— *Color histogram representation.* Color histogram is the most popular representation for appearance modeling in MOT systems as a result of its effectiveness to capture the statistical information of the target region. For example, Kratz and Nishino [2010] employ the color histogram model from [Pérez et al. 2002] to calculate the likelihood regarding the appearance, and they use a simple function like Eq. 2 to transform the histogram distance into probability. Similarly, to capture the dissimilarity, Sugimura et al. [2009] use the Bhattacharyya distance between hue-saturation color histograms when constructing a graph. In [Choi and Savarese 2010], for every object, a Mean-shift tracker employing the color histogram is utilized to sequentially seize the object. In [Leibe et al. 2008] appearance model is defined as the RGB color histogram of the trajectory. It is initialized as the first detection response's color histogram and evolves as a weighted mean of all the detection responses which are believed to belong to this trajectory. The likelihood considering the appearance is proportional to the Bhattacharyya coefficient of two histograms. Qin and Shelton [2012] deal with affinity considering appearance by calculating the Bhattacharyya distance between the average HSV color histograms of the concerned tracklets. Xing et al. [2009] represent the target (pedestrian body) as a constitution of three overlapped part areas, i.e. the Full Body (FB), the Head-Shoulder (HS), and the Head-Torso (HT). To link tracklets for data association, they consider the color histogram of each part in the detection response. The affinity regarding the appearance to link two tracklets T_i and T_j is calculated as in Eq. 3,

$$P_a(T_i, T_j) = \exp(-d(\mathbf{a}_i, \mathbf{a}_j)), \tag{3}$$

where $d(\mathbf{a}_i, \mathbf{a}_j) = mean\left\{d(\mathbf{a}_i^L, \mathbf{a}_j^L) | L = FB, HT, HS\right\}$, $\mathbf{a}$ means color histogram, and $d(\bullet, \bullet)$ is the Bhattacharya distance measure.
In [Zhang et al. 2008], the appearance term in the link affinity of two detection responses is based on RGB histograms. Given two detection observations $\mathbf{x}_i$ and $\mathbf{x}_j$, the corresponding color histograms $\mathbf{a}_i$ and $\mathbf{a}_j$ are extracted. Then the Bhattacharyya distance A_{ij} between $\mathbf{a}_i$ and $\mathbf{a}_j$ is obtained, and the probability of linking $\mathbf{x}_i$ and $\mathbf{x}_j$ is

$$P(\mathbf{a}_i | \mathbf{a}_j) = \frac{N(A_{ij}; A_s, \sigma_s^2)}{N(A_{ij}; A_s, \sigma_s^2) + N(A_{ij}; A_d, \sigma_d^2)}, \tag{4}$$

where $N(x; A_s, \sigma_s^2)$ and $N(x; A_d, \sigma_d^2)$ are two Gaussian distributions of A_{ij} between the same object and different objects respectively. The parameters of these two distributions are learned from the training data. Besides its advances, the color histogram representation has the drawback of losing spatial information.

— *Covariance matrix representation.* Covariance matrix is robust to illumination change, rotation, etc, so it is successful in the visual tracking problem. In [Henriques et al. 2011], the covariance matrix descriptor is employed to represent the appearance of an object. The likelihood concerning appearance to link two detection responses is modeled as

$$P_{link}(T_i, T_j) \propto N(d_{ij}^a, \Sigma_a), \tag{5}$$

Table III. An overview of appearance models employing multiple cues

References	Cues employed	Boosting	Parallel
[Kuo et al. 2010; Li et al. 2009] [Yang and Nevatia 2012c]	colors, HOG, covariance matrix, shapes	✓	×
[Song et al. 2010]	colors and foreground response	×	✓
[Mitzel et al. 2010]	colors and depth	×	✓
[Giebel et al. 2004] [Gavrila and Munder 2007]	shapes, colors and depth	×	✓
[Izadinia et al. 2012; Brendel et al. 2011]	colors, HOG, optical flow	×	✓
[Andriyenko and Schindler 2011]	HOG, Histogram of relative optical flow	×	✓
[Yang et al. 2009]	colors, shapes and bags of local features	×	✓
[Berclaz et al. 2006]	colors and occupancy map	×	✓

where $N(\bullet, \bullet)$ is the Gaussian distribution, d_{ij}^a is the appearance dissimilarity between T_i and T_j, and Σ_a is the covariance estimated from training data. In [Hu et al. 2012] a block-division appearance model which divides the region into blocks is proposed. Within each block, the covariance matrix is extracted as the region descriptor to characterize the block. At the same time, the likelihood of each block is computed with regard to the corresponding block of the target, and the likelihood of the whole region is the product of the likelihood of all the blocks.

— *Pixel comparison representation.* Nothing could be simpler than giving a binary result of comparison between two pixels, and this is the advance of this type of representation over other kinds of representation. Zhao et al. [2012] adopt Random Ferns in tracking similar to [Kalal et al. 2012]. It is actually to encode the results of comparisons between pairs of pixels and vote the comparison based on the training data. The probability of a patch to be positive is calculated based on how many positive samples and negative samples have been recorded by that leaf/code. The appearance probability is $\frac{s^+}{s^+ + s^-}$, where s^+ and s^- are the times have been recorded when the positive and negative samples drop in the leaf, respectively.

— *Bag of words representation.* Fast dense SIFT-like features [Lowe 2004] are computed in [Yang et al. 2009] and encoded based on the bag-of-word model. To incorporate the spatial information, the spatial pyramid matching (SPM) method [Lazebnik et al. 2006] is adapted. This is used as one observation model for appearance modeling.

3.2.3. Multi-cue based appearance model. Different kinds of cues could compensate each other, further make the appearance model robust. There arises an issue that how to fuse the information of multiple cues. Regarding this, we present this aspect based on two kinds of fusion strategies, *boosting* and *parallel* (also see Table III). The strategy of Boosting usually selects a few features from a feature pool sequentially via a Boosting based algorithm (e.g. Adaboost [Kuo et al. 2010], RealBoost [Yang and Nevatia 2012c]). The features have different discrimination power. On the contrary, parallel strategy regularly assumes cues are independent. The final affinity is the weighted summation or the product of affinities in terms of different cues.

— *Boosting.* A discriminative appearance model is proposed in [Kuo et al. 2010] to assign high similarity to tracklets which are of the same object, but low affinity to tracklets of different objects. Specifically, color histogram in RGB space, HOG and covariance matrix descriptor are employed as features. They choose 15 regions so that they have 45 cues in total in the feature pool. Collecting the positive and negative training pairs according to the so-called spatial-temporal constraints, they employ Adaboost to choose the most representative features to discriminate pairs of tracklets belonging to the same object from those belonging to different objects. Similarly, Yang and Nevatia [2012c] adopt features from [Kuo et al. 2010] and employ the standard RealBoost algorithm to learn the feature weights from training sample

set, which is composed of correctly linked pairs (as positive samples) and incorrectly matched pairs (as negative samples). A HybridBoost algorithm is proposed [Li et al. 2009] to automatically select features with maximum discrimination. This algorithm employes a hybrid loss function composed of a classification term and a ranking term. The ranking term ranks correct tracklet associations higher than their counterparts and the classification term dismiss wrong associations.

— *Parallel.* The Bhattacharyya distance between color histograms corresponding to the candidate rectangle region and the target initial region as well as the foreground response are utilized in [Song et al. 2010] to estimate the observation likelihood. In this work, these two cues are assumed to be independent. Mitzel et al. [2010] simultaneously segment and track multiple objects. As pedestrians and backgrounds often contain the same color, color information alone yields rather unreliable segmentation for pedestrians. To address it, they integrate the color information with depth information. Since the depth of the foreground is not constant, they firstly compute an expected depth of the foreground in the following frames according to the current depth and a maximum velocity. Then each depth of the foreground in the following frame could be assigned a probability based on a Gaussian distribution centered at the expected depth. The probability based on color representation is the accordance of color histogram calculated in the Lab space with the learned appearance model [Bibby and Reid 2008]. Then two probabilities computed from the color and depth cues are weighted by a parameter α as in Eq. 6. The similar weighting strategy is adopted in [Liu et al. 2012] to balance two cues of raw pixel intensity and silhouette.

$$P_i = (1-\alpha)P_{i,color} + \alpha P_{i,depth}, i \in \{f, b\}\,. \tag{6}$$

In [Izadinia et al. 2012], appearance information is considered in two stages. Detection responses are linked into short tracklets in the first stage. The affinity to link two detection responses is calculated according to color histogram of $9 \times 9 \times 9$ bins, a HOG feature descriptor for the entire bounding box enclosing the pedestrian, and a motion feature descriptor with optical flow binned into 12 intervals. In the second stage, a pedestrian-specific appearance model considers multiple parts of the object. Color information is employed to represent the object by concatenating the pixel level RGB descriptor of each part. A SVM model classifier is trained to distinguish a specific target from targets in its temporal window. Color, HOG and optical flow are concatenated and further processed with PCA projection for dimension reduction in [Brendel et al. 2011] to describe the detection response. The similarity w between two detection responses $\mathbf{z}$ and $\mathbf{z}'$ is

$$w = \exp\left(-\left(\mathbf{z}-\mathbf{z}'\right)^T \mathbf{M}\left(\mathbf{z}-\mathbf{z}'\right)\right), \tag{7}$$

where $\mathbf{M}$ is a distance metric matrix learned online.

Andriyenko and Schindler [2011] take the appearance from the detection responses into consideration by integrating an observation term into the energy function. HOG features and histograms of the relative optic flow [Walk et al. 2010] are used. A trajectory has lower energy if it overlaps regions of high pedestrian likelihood in the individual frames.

Another example [Yang et al. 2009] integrates multiple cues including colors, shapes and bags of local features based on [Lowe 2004; Lazebnik et al. 2006] as three different observation models to calculate the likelihood of linking a detection response to an existing trajectory. Assuming these three cues are $\mathbf{z}^1$, $\mathbf{z}^2$ and $\mathbf{z}^3$, the likelihood to link a detection response $\mathbf{x}$ to a trajectory is modeled as $P(\mathbf{z}^1, \mathbf{z}^2, \mathbf{z}^3|\mathbf{x})$. These cues are assumed to be independent from each other, thus the likelihood

is $P(\mathbf{z}^1, \mathbf{z}^2, \mathbf{z}^3|x) = \prod_{n=1}^{3} P(\mathbf{z}^i|\mathbf{x})$. This likelihood can be further modeled using a weighted linear combination of multiple dissimilarity function $d(\bullet)$ corresponding to multiple cues as

$$d(\mathbf{z}^1(x), \mathbf{z}^2(x), \mathbf{z}^3(x)) = \sum_{n=1}^{3} w_n d(\mathbf{z}^n(x)), \tag{8}$$

where $\mathbf{z}^i(x)$ is the image observation of the ith cue.

Three cues, shape, texture and depth, are integrated in a Bayesian framework to compute the weight of a particle [Giebel et al. 2004]. These cues are assumed to be conditionally independent. The shape similarity between the predicted shapes and the observed edges is measured by multi-feature distance transforms. The texture similarity is based on the Bhattacharyya distance between the reference texture distribution and the observed texture distribution. Similarity concerning depth is reversely proportional to difference between the measurement depth and the predicted distance.

Gavrila and Munder [2007] use cues of depth, shape and texture in a cascade manner to narrow the search space for multiple object detection and tracking. Specifically, depth is used to generate hypotheses, shape and texture cues are employed sequentially and finally the depth is utilized to verify the detections. Real-time performance is achieved by doing so.

Dividing the scene under multiple cameras into multiple grids, appearance model in [Berclaz et al. 2006] is constructed based on color model and ground plane occupancy estimation. The ground plane occupancy stands for the occupancy of an object at a specific grid. The color model is grid-wise, which is aware of the presence of an object. These two cues are also independent.

Colors, texture and motion are employed [Takala and Pietikainen 2007] to conduct multiple object tracking based on the foreground blobs from background subtraction.

3.3. Motion Model

Object motion model is also known as the dynamic model, which describes how an object moves. It is important for multiple object tracking since it can predict the potential position of objects in the future frames, reducing search space. In general, objects are assumed to move smoothly in the image scene (*cf*. the abrupt motion is a special case). The popular motion models employed by multiple object tracking are divided into two main classes.

3.3.1. Constant velocity motion models/linear motion models. As the name indicates, objects following these models are assumed to move with constant velocity. This is the easiest and the most popular model [Shafique et al. 2008; Yu et al. 2007]. The velocity of object in the next time is the same as the current velocity (added by process noise independently drawn from some types of distributions). For example, Breitenstein et al. [2009] employ a constant velocity motion model to propagate particles like this:

$$\begin{aligned} (x, y)_t &= (x, y)_{t-1} + (u, v)_{t-1} \bullet \Delta t + \epsilon_{x,y}, \\ (u, v)_t &= (u, v)_{t-1} + \epsilon_{u,v}, \end{aligned} \tag{9}$$

where (x, y) and (u, v) represent the 2D image position and the velocity respectively, $\epsilon_{x,y}$ and $\epsilon_{u,v}$ are noise variables drawn from Normal distributions with the mean of zero and variances to account for the previous states of the object. Specifically, the position variance $\sigma^2_{x,y}$ varies with the size of the object, and the velocity variance $\sigma^2_{u,v}$ is

inversely proportional to the number of successfully tracked frames. The more number of the successfully tracked frames, the smaller the variance, i.e., the less particles to spread.

The dynamic model in [Andriyenko and Schindler 2011] is also a constant velocity model. To be specific, an energy term which accounts for the differences between the velocities of one object in different time instants is formulated in Eq. 10 as,

$$E_{dyn}(\mathbf{X}) = \sum_{t=1}^{F-2} \sum_{i=1}^{N} \left\| \mathbf{v}_i^t - \mathbf{v}_i^{t+1} \right\|^2, \tag{10}$$

where $\mathbf{v}_i^t$ is the velocity of target i at time t. It is computed as the displacement between the object positions in two continuous frames. The first summation accounts for all the F frames in the sequence and the second summation accounts for all the N trajectories/objects. Intuitively, this term penalizes the difference between the velocity and forces the trajectories to be somehow smooth.

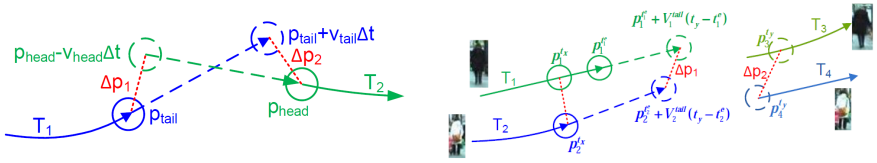
In [Xing et al. 2009], a constant velocity model simultaneously considering the forward velocity and the backward velocity is proposed to compute the affinity linking two tracklets in terms of motion. Given two tracklets T_i and T_j, let us assume there is a temporal gap between the tail of T_i and the head of T_j. The forward direction in the motion model is described by a Gaussian distribution centered in $\mathbf{p}_j^{head}$, the position of the head response in T_j, with variance Σ_j^B. It estimates the probability of the position of tail response in T_i plus some forward displacement $\mathbf{p}_i^{tail} + \mathbf{v}_i^F \Delta t$. Also the backward direction motion is also represented as a Gaussian, with the difference that motion is calculated backwardly from the position of head response in T_j to the position of tail response in T_i. The model is given in Eq. 11.

$$P_m(T_i, T_j) = G(\mathbf{p}_i^{tail} + \mathbf{v}_i^F \Delta t; \mathbf{p}_j^{head}, \Sigma_j^B) \bullet G(\mathbf{p}_j^{head} + \mathbf{v}_i^B \Delta t; \mathbf{p}_i^{tail}, \Sigma_i^F). \tag{11}$$

Different from previous graph based MOT approaches which treat each node as an individual observation (e.g, one detection response), Yang and Nevatia [2012b] employ the Conditional Random Field (CRF) model, treating the node in the model as a pair of tracklets. The label of each node indicates that whether the two tracklets corresponding to this node can be associated or not. To check whether these two tracklets could be linked or not, the unary term in the CRF model considers appearance and motion information. The probability to link them considering the motion is calculated based on the displacement between the estimated positions via a linear motion model and the real positions. Figure 7(a) from [Yang and Nevatia 2012b] illustrates this model clearly. Given two tracklets T_1 and T_2, assuming that T_1 is the front one along the time axis compared with T_2, there is a time gap Δt between the tail of T_1 and the head of T_2. The probability of linking T_1 and T_2 depends on two terms. One is from the displacement between the real position and the estimated position of the tail of T_1, which is defined as $\Delta \mathbf{p}_1 = \mathbf{p}^{head} - \mathbf{v}^{head} \Delta t - \mathbf{p}^{tail}$. The other one is from the displacement between the real position and the estimated position of the head of T_2, defined as $\Delta \mathbf{p}_2 = \mathbf{p}^{tail} + \mathbf{v}^{tail} \Delta t - \mathbf{p}^{head}$. The probability is

$$P_m(T_1, T_2) = G(\Delta \mathbf{p}_1, \Sigma_p) G(\Delta \mathbf{p}_2, \Sigma_p), \tag{12}$$

where $G(\bullet, \Sigma)$ is the Gaussian function with the mean of zero. This motion model is essentially the same as the one in [Xing et al. 2009]. It is quite popular [Kuo et al. 2010; Kuo and Nevatia 2011; Yang et al. 2011; Qin and Shelton 2012; Nillius et al. 2006]. However, it only considers the pair of tracklets itself. In [Yang and Nevatia 2012b],



(a) (b)

Fig. 7. The unary motion model (a) and pairwise motion model (b) from [Yang and Nevatia 2012b].

the motion model between two pairs of tracklets are also taken into consideration. Figure 7(b) from [Yang and Nevatia 2012b] shows the pairwise motion model. Considering two pairs of tracklets, T_1 and T_3, T_2 and T_4, they suppose T_1 and T_2 are tail-close tracklets. Firstly the earlier tail time when T_1 and T_2 end is defined as $t_x = min\{t_1^e, t_2^e\}$, where t_i^e means the ending time of tracklet T_i. Similarly, the later time when T_3 and T_4 start is represented as $t_y = max\{t_3^s, t_4^s\}$. Obviously $t_y > t_x$. Then they compute the relative distance between the estimated positions of T_1 and T_2 at frame t_y as $\Delta\mathbf{p}_1 = (\mathbf{p}_1^{t_1^e} + \mathbf{V}_1^{tail}(t_y - t_1^e)) - (\mathbf{p}_2^{t_2^e} + \mathbf{V}_2^{tail}(t_y - t_2^e))$, where $\mathbf{V}_1^{tail}$ and $\mathbf{V}_2^{tail}$ are the tail velocity of T_1 and T_2. On the other hand, the real relative distance between T_1 and T_2 at frame t_y is $\Delta\mathbf{p}_2 = \mathbf{p}_3^{t_y} - \mathbf{p}_4^{t_y}$. Similar to the motion model in the unary term, they also employ a zero-mean Gaussian function as $G(\Delta\mathbf{p}_1 - \Delta\mathbf{p}_2, \Sigma_p)$ to estimate the linkable probability. The insight behind this pair-wise motion model is that if the difference between $\Delta\mathbf{p}_1$ and $\Delta\mathbf{p}_2$ is small and T_1 is associated with T_3 (i.e., the label of the node corresponding to T_1 and T_3 is 1), then the probability to associate T_2 and T_4 is high.

Besides considering position and velocity, Kuo and Nevatia [2011] also take the accelerate rate into consideration. The probability concerning motion of a state $\{\hat{\mathbf{x}}_k\}$ (k is the frame index) given observation tracklet $\{\mathbf{x}_k\}$ (positions and sizes) is modeled as,

$$P(\{\hat{\mathbf{x}}_k\} \mid \{\mathbf{x}_k\}) = \prod_k G(\mathbf{x}_k - \hat{\mathbf{x}}_k, \Sigma_p) \prod_k G(\mathbf{v}_k, \Sigma_\mathbf{v}) \prod_k G(\mathbf{a}_k, \Sigma_\mathbf{a}), \tag{13}$$

where $\mathbf{v}_k = \frac{\hat{\mathbf{x}}_{k+1} - \hat{\mathbf{x}}_k}{t_{k+1} - t_k}$ is the velocity, $\mathbf{a}_k = \frac{\mathbf{v}_k - \mathbf{v}_{k-1}}{0.5(t_{k+1} - t_{k-1})}$ is the acceleration, and $G(\bullet, \Sigma)$ is the zero-mean Gaussian distribution.

3.3.2. Non-linear motion model. Linear motion model is commonly used to explain object's movement. However, there are some cases which the linear motion model cannot deal with. To this end, non-linear motion models are proposed to produce more accurate motion affinity between tracklets. For instance, Yang and Nevatia [2012a] employ the non-linear motion model to handle the situation that some targets may move freely. Given two tracklets T_1 and T_2 which belong to the same target in Figure 8(a), the linear motion model [Yang and Nevatia 2012b] would produce low probability to link them, which is not consistent with the truth. Alternatively, employing the nonlinear motion model, which is composed of a set of pattern tracklets M, the gap between the tail of tracklet T_1 and the head of tracklet T_2 could be reasonably explained by a tracklet $T_0 \in M$. As shown in Figure 8(b), the tracklet T_0 is a support tracklet to explain T_1 and T_2 because there exist responses $\{\mathbf{d}_i = (\mathbf{p}_i, \mathbf{s}_i, \mathbf{v}_i)\}$ in T_0 which are matched with the tail of T_1 and the head of T_2, where $\mathbf{p}$, $\mathbf{s}$ and $\mathbf{v}$ are position, size and velocity, respectively. Then the real path to bridge T_1 and T_2 is estimated based on T_0, and the similar way as the linear motion model is employed to calculate the affinity between T_1 and T_2, but based on the non-linear motion positions.

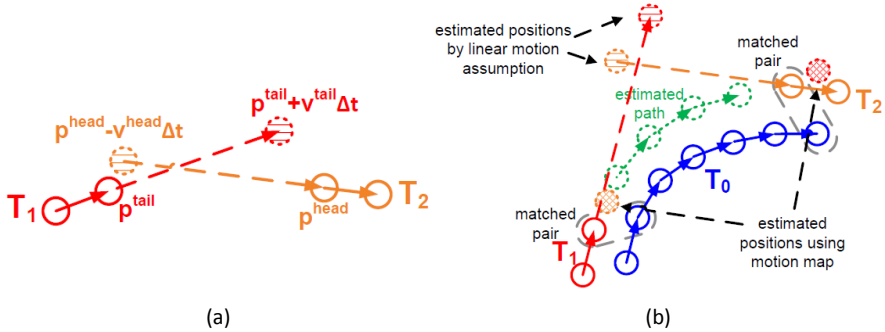


Fig. 8. An image from [Yang and Nevatia 2012a] to compare the linear motion model (a) with the non-linear motion model (b). Best viewed in color.

3.4. Interaction Model

The interaction model, also known as the mutual motion model, captures the influence of an object to other objects. This is a distinct issue of multiple object tracking compared with single object tracking. In the crowd scenery, obviously an object would consider some "force" from others. For instance, when a pedestrian is walking on the street, he would consider his speed, direction and destination, in order to avoid collision with others. Another example is that when a crowd of people walk across a street, each of them follows others and guides others at the same time, i.e., they form a motion pattern and every one follows this pattern. In fact, these two cases are examples of two typical interaction models known as the social force model [Helbing and Molnar 1995] and the crowd motion pattern model [Hu et al. 2008]. There are some representative work following these models, which are illustrated as follows.

3.4.1. Social force models. Social force models are also known as group models. These models do not assume objects to be independent, while each object is considered to be dependent from other objects and the environmental factors. This type of information could alleviate performance deterioration in crowded scenes. In social force models, targets are considered as agencies which determine their speed, velocity, destination based on the observations of other objects and the around environment. For instance, assuming that each pedestrian in a social group always adjusts its trajectory at an early stage in order to avoid possible collisions, Pellegrini et al. [2009] model this assumption by maximizing the minimum distance which makes the targets comfortable. In their model, each subject is represented as $s_i = \{\mathbf{p}_i^t, \mathbf{v}_i^t\}$, where $\mathbf{p}_i^t$ and $\mathbf{v}_i^t$ are the position and the velocity at time t. Regarding subject i, the minimum distance between it and subject j to make them comfortable is $d_{ij}^{*2}(\mathbf{v}_i)$. The corresponding energy term is E_{ij}. To model this property among multiple objects considering subject i, they balance each pair-wise energy by a weight. Thus, the final interaction model between subject i and other objects is

$$I_i(\mathbf{v}_i) = \sum_{r \neq i} w_r(i) E_{ir}(\mathbf{v}_i), \tag{14}$$

where $w_r(i)$ is the weight assigned to subject r considering subject i. Furthermore, they assume object i walks to a destination $\mathbf{z}_i$ with a desired speed u_i. Therefore they have two more energy terms as $S_i(\mathbf{v}_i) = (u_i - \|\mathbf{v}_i\|)^2$ penalizing sudden speed changes and $D_i(\mathbf{v}_i) = -\frac{(\mathbf{z}_i - \mathbf{p}_i)\cdot \mathbf{v}_i}{\|\mathbf{z}_i - \mathbf{p}_i\| \|\mathbf{v}_i\|}$ penalizing the drift from destination. Then the complete energy objective is

$$E_i(\mathbf{v}_i) = I_i(\mathbf{v}_i) + \lambda_1 S_i(\mathbf{v}_i) + \lambda_2 D_i(\mathbf{v}_i), \tag{15}$$

where λ_1 and λ_2 are parameters to balance the individual terms. By minimizing the above energy function for subject i, the search space of its destination could be largely reduced, and the data association procedure is further simplified.

Yamaguchi et al. [2011] also assume the objects are agencies of a social force model. The destination of an object is determined by considering the so-called personal, social and environmental factors, which are formulated as terms of an energy objective. In their model, each object is represented as $s_i = \{\mathbf{p}_i, \mathbf{v}_i, u_i, \mathbf{z}_i, A_i\}$ where $\mathbf{p}_i$ is the position, $\mathbf{v}_i$ is the velocity, u_i is the desired speed, $\mathbf{z}_i$ is the desired destination and A_i is the group of objects including object i. The behavior model to navigate objects has 6 terms, which are 1) a damping term penalizing the sudden change of velocity as $E_{damping}(\mathbf{v}; s_i) = |\mathbf{v} - \mathbf{v}_i|$, 2) a speed term giving a cost $E_{speed}(\mathbf{v}; s_i) = (u_i - |\mathbf{v}_i|)^2$ if the speed varies from the desired speed, 3) a direction term as $E_{direction}(\mathbf{v}; s_i) = -\frac{\mathbf{z}_i - \mathbf{p}_i}{|\mathbf{z}_i - \mathbf{p}_i|} \frac{\mathbf{v}}{|\mathbf{v}|}$ to cost if the object does not follow the desired direction, 4) an attraction term $E_{attraction}(\mathbf{v}; s_i, s_{A_i}) = \sum_{j \in \{A_i - i\}} (\frac{\mathbf{v}_i}{|\mathbf{v}_i|} \cdot \frac{\mathbf{v}_j}{|\mathbf{v}_j|})(\frac{\Delta \mathbf{p}_{ij}}{|\Delta \mathbf{p}_{ij}|} \cdot \frac{\mathbf{v}}{|\mathbf{v}|})$ assuming that people tend to stay close when they move together, 5) a group term $E_{group}(\mathbf{v}; s_i, s_{A_i}) = |\mathbf{v} - \bar{\mathbf{v}}_{A_i}|$ penalizing the variance of the velocity in a group ($\bar{\mathbf{v}}_{A_i}$ is the group average velocity) and 6) a collision term adopted from [Pellegrini et al. 2009] (see Eq.14 for details).

Qin and Shelton [2012] consider the social grouping behavior to improve data association for MOT. Besides the basic affinity model which traditionally accounts for the visual cues such as appearance and motion, they additionally take the social grouping behavior into consideration. To be specific, they assume people form K groups, where the parameter K can be learned optimally, and every tracklet assigned to the same group should be consistent with the group mean trajectory. Thus they have an extra cost term along with the traditional cost term from the basic affinity model as

$$\sum_{ik} \varphi_{ik} D(\tau_i, G_k), \tag{16}$$

where φ_{ik} is 1 if tracklet i is assigned to group k, and $D(\bullet, \bullet)$ is a distance measure concerning a tracklet and the group trajectory.

Two factors, repulsion and group motion, are considered in [Choi and Savarese 2010]. The repulsion factor tries to separate objects if they are too close to each other. Given two targets i and j at time t, the potential concerning the repulsion is $\varphi_r(i, j) = \exp(-\frac{1}{c_r r_{ij}})$, where r_{ij} is the distance between the two targets in the 3D space and c_r is a controlling parameter. It is obvious that if two objects are too close, then the potential between them is very small. Group motion factor assumes the relative distance between two objects in continuous two frame should keep unchanged. This also means that the velocities of the pair of objects should be similar to each other. Thus the group motion is modeled as a potential term $\varphi_g(i, j) = \frac{1}{1+\exp(s_g(r_{ij} - t_g))} \exp(-c_g \bullet \|\mathbf{v}_i - \mathbf{v}_j\|)$, where $\mathbf{v}$ is the velocity, c_g is a parameter to control this factor, the first factor is a soft step function to model how we consider two objects to be in the same group, t_g is a threshold distance and s_g is a parameter to balance the slope. From the group motion factor, if two objects are close enough to be considered in the same group, and they are with the similar velocity, then the potential is high.

A social force model which has four components is proposed in [Scovanner and Tappen 2009] to learn dynamics of pedestrians in the real world. These four components contribute four energy terms as 1) E_{LM} which limits the movement of a target to avoid the jump in the space grid, 2) E_{CV} which maintains the target's constant velocity, 3)

E_{Dest} which guides the target to reach a destination, 4) E_{AV} which accounts for other targets, producing repulsion in order to avoid the possible collisions. These four energy terms are weighted to form an energy objective which are then minimized to predict the movement of a target, generating tracks.

The data association problem and the group relationship mining problem are jointly estimated in [Pellegrini et al. 2010]. They model the trajectory assignment problem based on the motion and appearance information, and mine the group relationship by assuming the targets belonging to the same group keep the distance constant and have the same direction. A three-order CRF model is proposed and an energy function accounts for these two problems is constructed. By inferring the most probable estimation, these two problems are solved.

3.4.2. Crowd motion pattern models. Inspired by the crowd simulation literature, the motion patterns are introduced to alleviate the difficulty in tracking individual object in crowd. In general, this type of models is usually applied to the over-crowded scenario, i.e., the density of targets is considerably high. In the highly-crowded scenery, objects are usually quite small, and visual cues such as appearance and individual motion are ambiguous. In this case, the motion from the crowd is a comparably reliable cue for the problem.

There have been some work in this direction. For example, an assumption is made in [Ali and Shah 2008] that the behavior of an individual is determined by the scene layout and the surrounding objects. In order to model the influence from others and the scene structure, three kinds of force from the floor fields are proposed. These fields are Static Floor Fields (SFF), Boundary Floor Field (BFF) and Dynamic Floor Field (DFF). SEF accounts for the scene structure, including the favorite path the crowd takes and the sinking point (exit) of the scene. BFF takes the barriers in the scene into consideration by recognizing the physical and virtual barriers in the scene. DFF captures the motion of the crowd around the object being tracked to determine the future positions of objects in the crowd.

Zhao et al. [2012] deal with the multiple object tracking problem in the structured crowd scene by observing that the crowd exhibit clear motion patterns in this case, and these patterns could benefit the tracking problem. Motion patterns are discovered by the ND tensor voting [Mordohai and Medioni 2010] of the tracklet points which are obtained by KLT tracker. These motion patterns are represented as a set of 4-D points $\{\mathbf{q}_i = (x_i, y_i, v_{x_i}, v_{y_i}), i = 1, ..., n\}$, where x, y are spatial positions, and u, v are velocity along the two dimensions of the image. Then these motion patterns are employed to (1) estimate the probability of a candidate for detection and (2) predict the velocity of an object in a special position for tracking. (1) For detection, a candidate patch's probability of a detection is determined by the appearance as well as the motion patterns. Specifically, given a patch centered at (x_i, y_i), it has a set of correspondence points $\{\mathbf{p}_{ij}, j = 1, ..., m\}$ in the next frame. Each correspondence's probability concerning the motion patterns is voted by checking whether it is supported by the motion priors (n points) of (x_i, y_i), as

$$vote_{ij} = \sum_{l=1}^{n} \exp(-\frac{\left\|\mathbf{q}_l - \mathbf{p}_{ij}\right\|^2}{\sigma_1^2}). \tag{17}$$

Then these votes are normalized over all the m correspondences of point (x_i, y_i). (2) For tracking, given a point $\mathbf{p}_i = (x_i, y_i)$, its velocity is given by considering the appearance features and the motion patterns. Considering a square area around the concerned point, the weighted sum of the motion priors of each point in this area is used to give a velocity prior $\mathbf{v}_0$ of this point along with the covariance matrix Σ_v. To be more

reasonable, these points are weighted by a Gaussian kernel to produce the estimated velocity. After that, a cost term based on the Mahalanobis distance as

$$\ell_{MP}(\mathbf{v}) = (\mathbf{v} - \mathbf{v}_0)^T \Sigma_v^{-1} (\mathbf{v} - \mathbf{v}_0) \tag{18}$$

is integrated to the cost function to derive the final velocity v.

Observing that the group of pedestrians exhibits collective spatio-temporal structure, the movement of an object within any local space-time location of the video are learned by training a set of Hidden Markov Models (HMM) [Kratz and Nishino 2010; 2012]. The entire video is viewed as 3D volume, and it is divided into local spatio-temporal cubes. The motion pattern of a specific spatio-temporal cube is represented as a 3D Gaussian distribution considering the 3D gradients of all the pixels in the cube. This motion pattern is assumed to vary through the time and exhibit the Markov property. Thus the future motion pattern could be predicted based on the previous states (motion patterns), and this predict motion pattern could be employed to constraint the tracking of the object in this spatio-temporal location.

The motion pattern models described above make an assumption that the objects move coherently in a common direction. This may be hold in the case of the so-called structured crowd scenarios, but does not comply with the unstructured crowd which exhibits various modalities of motion. To address it, Correlated Topic Model (CTM) is adopted [Rodriguez et al. 2009] to learn the various motion behaviors in the scene. A tracker which can predict a rough displacement based on scene codebook from all the moving pixel in the scene, along with the learned high-level behavior, are weighted to track objects in the unstructured scenes. Similar to image retrieval, motion pattern could also be retrieved [Rodriguez et al. 2011]. Motion patterns are firstly learned in a unsupervised and off-line manner from a database composed of a large number of videos. Then given a test video, a set of space-time patches are matched to explain the test video. After that, the motion priors in the retrieved video patches are transferred to the test video as a prior to assist object tracking along with a Kalman filter based on tracker.

3.5. Exclusion Model

Exclusion is an obvious constraint when seeking a solution to the MOT problem due to the physical collisions. Given multiple detection responses and multiple trajectory hypotheses, generally there are two constraints to be considered. The first one is the so-called detection-level exclusion [Milan et al. 2013], i.e., two different detection responses in the same time instant (i.e. the same frame) cannot be assigned to the same trajectory hypothesis. The second one is the so-called trajectory-level exclusion, i.e., two trajectories cannot occupy the same detection response. Modeling of them is presented as follows.

3.5.1. Detection-level exclusion modeling. Milan et al. [2013] explicitly model the detection-level exclusion by defining a cost term to penalize the case if two simultaneous detection response d_i^t and d_j^t at time t are assigned the same label of trajectory with the cost $\bar{\varphi_X}$ if they are distant.

KC and De Vleeschouwer [2013] employ the label propagation for multiple object tracking. They construct multiple graphs to model the spatial-temporal relationship, appearance, exclusion, and propagate labels in the graphs. In order to model the exclusion, a special exclusion graph is constructed. Given all the detection responses in the image sequence, the exclusion graph captures the constraint that the detection responses with the same time stamp (occurring at the same time) should have different labels. They define a graph where the nodes represent detection responses. These

nodes are not fully connected. Alternatively, each node (one detection) is only connected to the nodes (other detections) happening at the same time as the node concerned. The connecting weight of each edge between two nodes are uniform as $W_{ij} = 1/n$, where n is the number of detection responses happening at the same time as these two nodes. After constructing this graph, the Laplacian of this exclusion graph could be computed as $L^{(-)}$ and the label error regarding the exclusion is maximized as,

$$\arg\max_{Y \in \mathcal{P}} Tr(Y L^{(-)} Y), \tag{19}$$

where $Y = (\mathbf{y}_1, ..., \mathbf{y}_{|V|})$ is the label assignment of all the $|V|$ nodes in the graph, $\mathcal{P}$ is the set of all row-stochastic matrices of size $|V| \times |V|$, $Tr(\bullet)$ is the trace norm of a matrix.

3.5.2. Trajectory-level exclusion modeling. In [Andriyenko and Schindler 2011], a term in Eq. 20 to model the exclusion is accounted in an energy function. It is

$$E_{exc}(\mathbf{X}) = \sum_{t=1}^{F} \sum_{i \neq j} \frac{s_g}{\left\| \mathbf{x}_i^t - \mathbf{x}_j^t \right\|^2}, \tag{20}$$

where $\mathbf{X}$ is the ground plane coordinates of all objects in all frames. The first summation accounts for all the F frames in the sequence, and the second summation accounts for all the detection responses in the same frame. $\mathbf{x}_i^t$ is the ground plane coordinate of object i at time t, and s_g is a scale factor. If two detection responses are too close, it will lead to great cost of the energy function.

To model the trajectory level exclusion, Milan et al. [2013] penalize the case that two close trajectories T_i and T_j have different labels. The penalty $h_X(T_i, T_j, f)$ considering the label f is proportional to the spatial-temporal overlap between T_i and T_j as $\varsigma(T_i, T_j) = \sum_{t \in O(T_i, T_j)} \varsigma_{i,j}^t$, where $O(\bullet, \bullet)$ is the common lifespan of two trajectories, and $\varsigma_{i,j}^t$ is the mutual overlap. The closer the two trajectories, the higher penalty it is.

Similarly, mutual exclusion is modeled as an additional cost term to penalize the case that two trajectories are very close to each other. This cost of this term is reversely proportional to the minimum distance between the trajectories in their temporal overlap. The cost term would abandon one of the trajectory to avoid the collision [Andriyenko et al. 2012].

In [Butt and Collins 2013], the exclusion is modeled as an extra constraint in the objective function of network flow. Let the detection observations at frame k as $F_k = \left\{ ob_1^k, ..., ob_{r_k}^k \right\}, k = 1, ..., l$, where l is the number of frames. Given detection responses in two consecutive frames as F_k and F_{k+1}, one detection from F_k as $ob1_{m_i^k}$ and one detection from F_{k+1} as $ob2_{m_i^{k+1}}$ can form a match as m_i^k. They represent all the matches between these two frames as $P_k = \left\{ m_1^k, ..., m_{n_k}^k \right\}$. Based on this, a graph is constructed as $G = (V, E)$, where each node in G is a pair of detections and each edge belonging to E represents the flow in the graph, where flow 1 means linkable and 0 means not. Recalling the constraint that one detection should only be occupied by no more than one trajectory, for each of the matches $(a, *)$ in P_k having a common node a as the incoming node, edges from P_{k-1} entering node a should obey the constraint that only one can be 1 and the other should be 0. These edges are called conflict edges. Similarly, for match pair $(*, a)$ in P_{k-1} having a as the outgoing node, conflict edges are those exiting node a. All the conflict edges are represented as $EC_1, ..., EC_q$. They limit the sum of the flow to be at most 1 as an constraint of the min-cost network flow optimization function in Eq. 21 (we here ignore the flow balance constraint for simplicity),

0.5533	0.2974	0.7509	0.3832
0.4002	0.2475	0.4501	0.5745
1.4403	0.5524	0.2771	1.4751
2.5272	3.0886	3.6015	5.4790

Fig. 9. An image from [Hu et al. 2012] to illustrate how the occlusion is handled. Left: one object is occluded by another object. Right: the reconstruction error map of the occluded object

$$\min f(x) = \sum_{(i,j)\in E} c_{ij}x_{ij}, \quad s.t. \sum_{(i,j)\in EC_s} x_{ij} \leq 1, s = 1, ..., q \tag{21}$$

where c_{ij} is the cost to link two pairs considering appearance and path smoothness constraints, $x_{ij} \in \{0, 1\}$ is the flow.

3.6. Occlusion Handling

Occlusion is a fatal issue to distinguish multiple object tracking from single object tracking as it is frequent in MOT. When occlusion happens in CFT, it is difficult to utilize the appearance information. Thus, when objects involved in occlusion, it is confusing to determine which track that each object belongs to. Occlusion in ABT also leads to problems like ID switch in data association for multiple possible assignments.

Hu et al. [2012] propose a block-division model to deal with occlusion along with the task of recovering occlusion relationship among objects in CFT. In this model, the object is divided into multiple un-overlapped blocks and for each block an appearance model based on subspace learning is constructed. The likelihood is computed according to the reconstruction error in the subspace corresponding to each block. This model benefits the tracking problem under occlusion in two aspects. The first one is that as the object is divided into blocks and the likelihood of one candidate is the product of all the likelihood of each block, the spatial information is accounted for. The second one lies in that an occlusion map could be obtained according to the reconstruction errors of all the blocks, and this occlusion map can determine which object is occluded and furthermore which part of the object is occluded. This information is very helpful since based on this the appearance model could be selectively updated, making the appearance model up to time. One image from [Hu et al. 2012] could be employed to illustrate this. As shown in Figure 9, occlusion happens between two objects, and the reconstruction errors corresponding to all the blocks of the occluded object are shown in the right. It is obvious that, the reconstruction error of the occluded blocks (the bottom part of the girl's face) is clearly higher than those of the unoccluded blocks. Based on this, the occlusion relationship among objects can be recovered and it further helps to keep the model up to time.

An Explicit Occlusion Model (EOM) is proposed in [Zhang et al. 2008] and integrated into the cost-flow framework to better handle long-term occlusion in data association for MOT. In the classical data association, two tracklets are assumed to be linkable only when the temporal gap between them is small. This would somehow result in fragments in the final association estimation. Increasing the temporal gap threshold to make more tracklets linkable could ease the situation to some extent, but would possibly yield more errant associations. To deal with this, occlusion hypotheses are generated based on the occlusion constraints. If the distance and scale difference of two observations are small enough, then they are occludable. Assuming $\mathbf{x}_i$ is occluded by $\mathbf{x}_j$, the occlusion hypothesis is $\widetilde{\mathbf{x}}_i^j = (x_j, s_i, a_i, t_j)$, where x_j and t_j are the position and

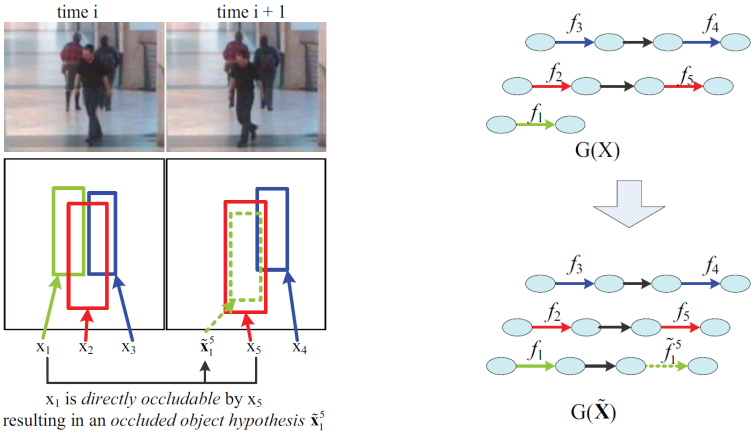


Fig. 10. The occlusion hypothesis illustration from [Zhang et al. 2008]. The solid rectangles are input observations and the dashed-line rectangle is an occluded object hypothesis.

time stamp of $\mathbf{x}_j$, and s_i and a_i are the size and appearance of $\mathbf{x}_i$. An image from [Zhang et al. 2008] is shown in Figure 10. $\mathbf{x}_1$ is occluded by $\mathbf{x}_5$, and an occlusion hypothesis $\tilde{\mathbf{x}}_1^5$ is generated. Along with the original observations (tracklets), all these observations are given as input to the cost-flow framework and the MAP is conducted to obtain the optimal solution.

Instead of the strategy of hypothesize-and-test for occlusion handling, Ryoo and Aggarwal [2008] propose the observe-and-explain strategy to handle the inter-object occlusion and scene-object occlusion. Their strategy could save computation cost as an observation mode is activated when the state of tracking is not clear due to occlusion. When they get enough observations, explanations are generated to correspond to the observations.

Part based model is a popular choice to tackle occlusion. For example, in [Yang and Nevatia 2012c], part-based appearance model is learned to better discriminate an object from other objects around and the background. To explicitly deal with occlusion, the object is represented as 15 parts. Given a tracklet T_k, its appearance model consists of a set of features $F_k = \{f_k^1, f_k^2, ..., f_k^n\}$ and the corresponding weights $W_k = \{w^1, w^2, ..., w^n\}$ from its head observation and its tail observation. The link probability considering appearance between two tracklets is calculated as the similarity between the tail of one tracklet and the head of the other tracklet, which is $\sum_i w_i h_i(f_j^i, f_k^i)$, where f_j^i and f_k^i are from the ith part of different objects/tracklets and $h(\bullet, \bullet)$ is a similarity evaluation function. The meaning of the part model is that once a part is found occluded, all the features from that part are assumed to be invalid. The learning of the appearance model is conducted via a boosting algorithm.

Part based model is also applied in [Izadinia et al. 2012] as multi-person multi-part tracker. Beginning from a state-of-the-art part-based human detector [Felzenszwalb et al. 2010], they track the whole human body and the individual body parts, and the final trajectory estimation is obtained by jointly considering the association between the whole human body and the individual human body parts. Figure 11 shows how the part based model handles occlusion. The pedestrian is occluded from frame 47 to frame 134. During this period, the whole-body human detector would be confused. However, thanks to the part detector, the visible parts are detected. Based on these parts, the trajectories of the visible part are estimated. Furthermore, along with the trajectory of the whole body, the complete trajectory is recovered.

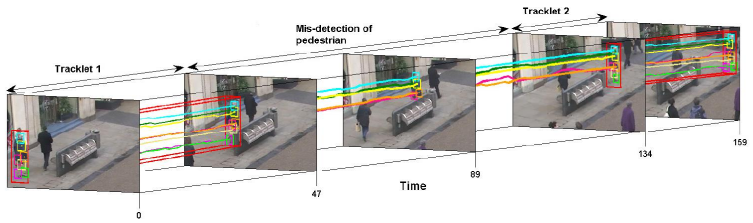


Fig. 11. An image from [Izadinia et al. 2012] to illustrate how the part based model deals with occlusion.

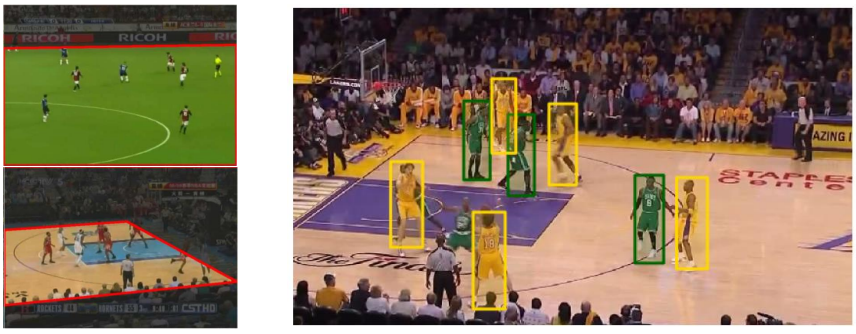
Fig. 12. Training samples for the double-person detector in [Tang et al. 2014]. From left to right, the level of occlusion increases.

As occlusion results in interruption in the tracks, Brendel et al. [2011] aim to deal with it progressively based on the proposed MWIS algorithm. Starting from a graph within which each node represents a pair of tracks, weight of the node is the affinity of the pair of tracks, and edge is connected if two nodes share common detections, MWIS algorithm outputs a new graph containing longer tracks. This procedure is iterated until convergence (output result does not change).

The model adopted in [Tang et al. 2013; Tang et al. 2014] is like a split-and-merge fashion to handle occlusion. As is known, severe occlusion may lead to failure of detectors. Different from the traditional detector which treats occlusion as distraction, occlusion is employed to help detection by observing that occlusion yields typical appearance patterns. Specifically, a double-person detector is built to handle different levels of occlusion between two people. They train the double-person detector based on the instances generated by synthetically combining two objects with different levels of occlusion, thus the resulting detector can be occlusion-aware. Figure 12 shows the training instances for the double-person detector. Along with the traditional single person detector, this multi-person detector could be employed as the basis of multiple object tracking.

In general, tracking based on appearance information may fail when occlusion happens. Analogous to the part based model which can still observe some parts in case of occlusion, feature point clustering based tracking, which assumes feature points with similar motion should belong to the same object, is also applicable to handle occlusion. As long as some part of the object is visible, the clustering of feature point trajectories will work. Examples can be found in [Sugimura et al. 2009; Brostow and Cipolla 2006; Fragkiadaki et al. 2012].

Mitzel et al. [2010] combine a level-set tracker based on image segmentation and a high-level tracker based on detection for MOT. In their approach, the high-level tracker is employed to initialize new tracks from detection response and the level-set tracker is used to tackle the frame-to-frame data association. When occlusion occurs, the level-set tracker would fail. To handle this, the high-level tracker keeps a trajectory alive for up to 15 frames when occlusion happens, and extrapolates the position to grow the dormant trajectory through the occlusion. In case the object reappears, the track is fired again and the identity is maintained. The similar idea is also used in [Mitzel and Leibe 2011].



(a) (b)

Fig. 13. (a) Images from [Xing et al. 2011] show results of field segmentation and (b) images from [Lu et al. 2013] show results of team classification.

4. MOT SPECIAL CASES

Some publications cannot be simply grouped into traditional multi-target tracking as they exhibit different attributes. We represent three special cases as follows.

4.1. MOT in Sport Scenarios

Tracking of multiple targets in the sport court has wide applications such as the technical statistics in the match, automatic analysis of strategies. There are several differences between traditional MOT and this special problem.

The first one is that the sport court is special. There are frequent shot switches and actions to zoom in and zoom out, which resut in challenges to this problem. However, this scene also exhibits useful features. For example, regarding some kinds of sport such as basketball and football, there is a clear boundary between the court and the background. This kind of boundary could dismiss most of the confusion from the background once it is recognized. For instance, Xing et al. [2011] propose to segment the playfield from the nonplayfield as a pre-processing for MOT. The segmentor consists of an offline stage and an online stage using multiple useful cues, including color, motion, and shape. Figure 13(a) shows some results of field segmentation from [Xing et al. 2011].

The second difference is that targets in this problem probably dress uniform. This can lead to two effects. On one hand, as the targets dress very similarly to each other, the appearance of targets is not as diverse as traditional MOT. Thus it may be more difficult to differentiate objects. On the other hand, as claimed before, targets of the same team wear clothes of the same color or pattern, while ones from different teams do not follow this. It could be helpful. For example, Lu et al. [2013] train a logistic regression classifier which maps image patches to team labels using RGB color histograms. Figure 13(b) shows team classification result of a basketball match between the team of Celtic and Lakers. This step improves the precision while retaining the same level of recall rate.

4.2. MOT in Aerial Scenes

Multiple object tracking in aerial scenes is different from the normal MOT as it exhibits these features: 1) the size of targets is considerably small, thus the commonly used appearance information is not reliable, 2) the frame rate is quite low, 3) only the gray-scale appearance information is available, 4) very high density, the number of objects could be up to hundreds of. Therefore, the solution is also different from that

to the normal multiple object tracking problem. In the following, some examples about this special problem are described.

Reilly et al. [2010] propose several techniques to tackle the difficulties mentioned above. To detect objects, motion detection is applied after background modeling. Obtaining objects in each frame, Hungarian algorithm is adapted to solve the assignment of detections to existing trajectories. When computing the affinity, the spatial proximity, velocity orientation and context are considered. Note that context is relatively important in this special case.

Shi et al. [2013] deal with the multi-frame multi-target association problem in the aerial scene as a rank-1 tensor approximation problem. High-order tensor is constructed based on all the candidate trajectories. Then the multi-dimensional assignment problem for MOT is transformed into the rank-1 approximation. In the end the approximation is solved by a l_1 tensor power iteration.

4.3. Generic MOT

As we have discussed, the majority of the MOT work focuses on some special kinds of objects, such as the vehicles, pedestrians. The reason is that ABT solutions without manual initialization are more practical in real-life applications and object detection has achieved great progress, especially for pedestrians. However, these methods, although free of manual initialization, still rely on special kind of object detectors to obtain detection observations. These object detectors are typically trained already in advance. There arises two problems that, 1) these pre-trained detectors can only be applied to the image sequences of the same type of objects, 2) as these detectors are pre-trained, they are data-specific. Thus the performance is not the optimal when they are applied to other image sequences. On the other hand, the CFT approaches, although require manual initialization, they can be applied to type-free sequences.

Observing this, recently there is a trend to generalize the MOT problem to any kind of objects, free of off-line trained detectors and free of manual initialization. It overcomes drawbacks of both ABT and CFT methods. Some researchers have proceeded a step to investigate this problem.

Zhao et al. [2012], propose to track multiple similar objects by requiring one instance in the first frame to be labeled. They firstly track this object, collect training samples, and train an object detector for this kind of objects in the first a few frames. Then they start from the first frame again, detect the top N (specified by the user) similar objects and track them in the subsequent frames, treating these objects as independent objects. The difference between [Zhao et al. 2012] and CFT is that [Zhao et al. 2012] saves much labeling labor compared with CFT. However, as the number of objects to track is still fixed, it is not comparable with ABT.

To make the generalization of MOT more practical, Luo and Kim [2013] propose the generic MOT problem and solve it by multiple task learning [Evgeniou and Pontil 2004; Caruana 1997]. In the generic MOT problem, objects are similar to each other regarding the appearance. For initialization, it also requires labeling of one example in the first frame. Then the training samples for object detection are collected and accumulated, thus the object detector is progressively refined. A specialized tracker will be initialized once an object is discovered. Since performance is improved by learning related tasks simultaneously, tracking of these multiple objects are handled by multiple task learning.

Dealing with the same problem as that in [Luo and Kim 2013], Luo et al. [2014] propose a bi-label propagation framework composed of three layers, layer of sliding windows, layer of detection responses and layer of trajectories. Each sliding window in this framework is associated with bi-labels, one is class label which indicates whether it is an object or the background, and the other one is object label, which represents

its identity. As one initial bounding box is given in the first frame, the detection and tracking of multiple objects is viewed as the class and object label propagation in the spatio-temporal video tube, which are integrated in a framework from a Bayesian perspective and verified by experiments on several data sets.

Regarding that one object shares appearance similarity with other objects, multiple similar objects are tracked in [Dicle et al. 2013]. However, different from [Luo and Kim 2013; Luo et al. 2014], the detection responses are given as input to the algorithm. They utilize the motion information for tracklet association.

[Brostow and Cipolla 2006] is an example which deals with generic MOT without any supervision. Assuming that a pair of points that appears to move together is likely to be part of the same individual, feature points are tracked and motion clustering is conducted to discover and maintain identical entities.

5. MOT EVALUATION

Given a MOT algorithm developed, we need metrics, data sets to evaluate its performance. Also, it is important to compare a developed solution with resonable baseline algorithms. In the following, we list the metrics, publicly available data sets and baseline algorithms.

5.1. Metrics

Evaluation metrics of MOT approaches are crucial as they provide standard for fair quantitative comparisons. A brief review on different MOT evaluation metrics is presented. As many approaches for MOT employ the tracking-by-detection strategy, they often measure the detection performance as well as tracking performance. Some metrics for object detection are therefore employed in MOT approaches. Based on this, MOT metrics can be largely categorized into two sets evaluating detection and tracking respectively, as listed in Table IV.

5.1.1. Metrics for detection. We further group metrics for detection into two subsets. One set measures accuracy of the result, and the other one measures the precision.

— *Accuracy.* The commonly used Recall and Precision metrics as well as the average False Alarms per Frame (FAF) rate are employed as MOT metrics [Yang et al. 2011]. Choi and Savarese [2010] use the False Positive Per Image (FPPI) to evaluate the detection performance in MOT. A comprehensive metric called Multiple Object Detection Accuracy (MODA), which accounts for the relative number of false positives and miss detections is utilized [Kasturi et al. 2009].
— *Precision.* The Multiple Object Detection Precision (MODP) metric measures the quality of alignment between the true detection and the ground truth [Kasturi et al. 2009].

5.1.2. Metrics for tracking. Metrics for tracking are classified into four subsets by different attributes as the following.

— *Accuracy.* This kind of metrics measures how accurately an algorithm could track the target objects. The metric of ID switches (IDs) [Yamaguchi et al. 2011] counts how many times the MOT algorithm switches to the wrong objects. Multiple Object Tracking Accuracy (MOTA) metric [Keni and Rainer 2008] combines the false positive rate, false negative rate and mismatch rate for MOT.
— *Precision.* Two metrics, Multiple Object Tracking Precision (MOTP) [Keni and Rainer 2008] and Tracking Distance Error (TDE) [Kratz and Nishino 2010] belong to this subset. They describe how precisely the objects are tracked from the view of overlap and distance.

Table IV. An overview of evaluation metrics for MOT. The up arrow (*resp.* down arrow) indicates that the performance is better if the quantity is greater (*resp.* smaller)

Type	Concern	Metric	Description	Note
Detection	Accuracy	Recall	correctly matched detections over ground-truth detections	↑
		Precision	correctly matched detections over result detections	↑
		FAF/FPPI	number of the false alarms averaged over the sequence	↓
		MODA	take the miss detection, the false positive rate into account	↑
	Precision	MODP	the overlap between the true positives and the ground truth	↑
Tracking	Accuracy	MOTA	take the false negative, false positive and mismatch rate into account	↑
		IDS	the number of times that a tracked trajectory changes its matched ground-truth identity	↓
	Precision	MOTP	overlap between the estimated positions and the ground truth averaged over the matches	↑
		TDE	difference between the ground-truth annotation and the tracking result	↓
	Completeness	MT	percentage of ground-truth trajectories which are covered by tracker output for more than 80% in length	↑
		ML	percentage of ground-truth trajectories which are covered by tracker output for less than 20% in length	↓
		PT	1.0 - MT - ML	-
		FM	the number of times that a ground-truth trajectory is interrupted in tracking result	↓
	Robustness	RS	the ratio of the tracks which are correctly recovered from short occlusion	↑
		RL	the ratio of the tracks which are correctly recovered from long occlusion	↑

— *Completeness.* Metrics for completeness indicate how completely the ground truth trajectories are tracked. Mostly Tracked (MT), Partly Tracked (PT), Mostly Lost (ML) and Fragmentation (FM) [Li et al. 2009] could be grouped to this set.
— *Robustness.* To assess the ability of a MOT algorithm to recover from occlusion, metrics of Recover from Short-term occlusion (RS) and Recover from Long-term occlusion (RL) are considered [Song et al. 2010].

5.2. Data Sets

To compare with various state-of-the-art MOT methods, publicly available data sets are employed to evaluate the proposed methods in individual publications. We here summarize the popular data sets in the literature, to give a clear view in Table V.

5.3. Public Algorithms

We examine the literature and list the algorithms of which their associated source codes are publicly available to make further comparisons convenient in Table VI.

6. CONCLUSION AND FUTURE DIRECTIONS

This paper has presented a comprehensive review of Multiple Object Tracking (MOT). The review is done in the following way of developing a MOT system: the current state of this topic, all the key issues under consideration, evaluation metrics and data sets. The state of the topic is discussed by its scenarios and approach categorization. As for the key aspects of MOT approaches, we have given the recent developments in details with classification and case study. Some popularly employed metrics and data sets, along with public algorithms are listed for evaluation of MOT approaches.

Although great progress in MOT has been recently achieved, there are still remaining issues to be tackled.

Table V. An overview of publicly available data sets. The tick means ground truth is available while the cross means not available

Item	Multi-view	Ground truth	Crowd degree[a]	Web link
PETS 2009	✓	✓	SC	www.cvg.rdg.ac.uk/PETS2009/a.html
PETS 2006	✓	×	SC	www.cvg.rdg.ac.uk/PETS2006/data.html
PETS 2007	✓	✓	SC	www.pets2007.net/
CAVIAR[b]	✓	✓	UC	http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/
Trecvid 2008	✓	×	SC	www-nlpir.nist.gov/projects/tv2008/
TUD	×	✓	SC	www.d2.mpi-inf.mpg.de/datasets
Caltech Pedestrian	×	✓	UC	www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/
UBC Hockey	×	×	SC	www.cs.ubc.ca/~okumak/research.html
Lids AVSS 2007	×	✓	SC	www.eecs.qmul.ac.uk/~andrea/avss2007_d.html
ETH pedestrian	✓	✓	SC	www.vision.ee.ethz.ch/~aess/dataset/
ETHZ Central	×	✓	SC	www.vision.ee.ethz.ch/datasets/
Town Centre	×	✓	SC	www.robots.ox.ac.uk/ActiveVision/Research/Projects/2009bbenfold_headpose/project.html#datasets
Zara	×	×	UC	https://graphics.cs.ucy.ac.cy/research/downloads/crowd-data
UCSD	×	×	SC	http://www.svcl.ucsd.edu/projects/anomaly/dataset.htm
UCF Crowds	×	×	OC	www.crcv.ucf.edu/data/crowd.php

[a]The density is assessed by human perception. UC: under-crowded, SC: semi-crowded and OC: over-crowded.
[b]Part of the data is recorded by only one camera.

Table VI. List of publicly available program codes

Item No.	Reference	Web Link
1	[Choi and Savarese 2010]	https://www.eecs.umich.edu/vision/mttproject.html
2	[Jiang et al. 2007]	http://www.cs.bc.edu/~hjiang/details/tracking/index.html
3	[Andriyenko and Schindler 2011] [Milan et al. 2014]	http://research.milanton.de/contracking/
4	[Andriyenko et al. 2012]	http://research.milanton.de/dctracking/
5	[Zamir et al. 2012]	http://crcv.ucf.edu/projects/GMCP-Tracker/
6	[Berclaz et al. 2011]	http://cvlab.epfl.ch/software/ksp
7	[Okuma et al. 2004]	http://www.cs.ubc.ca/~okumak/research.html
8	[Zhang and van der Maaten 2013b] [Zhang and van der Maaten 2013a]	http://visionlab.tudelft.nl/spot
9	[Pirsiavash et al. 2011]	http://www.ics.uci.edu/~dramanan/
10	[Rodriguez et al. 2009]	http://www.mikelrodriguez.com/crowd-tracking-matlab-application

— *MOT with video adaptation.* As aforementioned, the majority of current MOT methods is ABT which requires an offline trained object detector. There arises a problem that the detection result is not optimal since the object detector is not trained for a given specific video. This further decreases the performance of multiple object tracking. The customization of the object detector is necessary to improve the performance. One solution is proposed in [Shu et al. 2013], which adapts a generic pedestrian detector to a specific video by progressively refining the generic pedestrian detector. This is an important direction to improve the pre-procedure for MOT.

— *Balance between crowd density and completeness of object representation.* This issue involves the scenarios described in Sec. 2.2. In general, the denser the crowd is, the less ratio of the object's body is visible. In the under-crowd scenery, the whole body of the object can be recovered easily. However, in the over-crowd scenery, often only the head of the object is observable. With the minimum apperance information given, the motion pattern of the crowd could be, alternatively, helpful for MOT since targets exhibit coherent motion patterns.

— *MOT under multiple cameras.* It is obvious that MOT would benefit from multi-camera settings [Hofmann et al. 2013; Chang et al. 2000]. There are two kinds of configurations of multiple cameras. The first one is that multiple cameras record the same scene, i.e., multiple views. How to fuse information from multiple cameras is

a key issue in this configuration. The second one is that each camera of multiple camera records a different scene, i.e., a multi-camera network. One issue in the data association of multiple cameras is the object re-identification problem.

— *Multiple 3D object tracking.* Most of the current approaches focus on multiple object tracking in 2D, even in the cases of multiple cameras. 3D tracking [Park et al. 2008], which could provide more accurate position, size estimation and effective occlusion handling for high-level computer vision tasks, is potentially more useful. However, 3D tracking requires more parameters to estimate and more computation cost compared with 2D tracking. Meanwhile, 3D model acquisition is another issue exclusive to 2D MOT.

— *MOT with scene understanding.* Ever more attention is paid to the over-crowded scenarios such as underground station, train station and so on. In this kind of scenario, most of the objects are very small and/or largely occluded. To this end, scene understanding, including crowd analysis [Rodriguez et al. 2011; Rodriguez et al. 2009; Zhou et al. 2012; Zhou et al. 2012] for the objects themselves and scene structure recognition (exit,entrance, etc) [Zhou et al. 2011] is being investigated to help multiple object tracking in this scene.

— *MOT with other CV tasks.* Though multiple object tracking is in serve of other high-level computer vision tasks, there is a trend to solve multi-object tracking as a mid-level computer vision task and some other computer vision tasks at the same time as they are beneficial to each other. For example, Zhang et al. [2007] tackle multiple object tracking and pose estimation simultaneously. Detection and tracking are combined and better results are achieved in [Andriluka et al. 2008]. Ishiguro et al. [2008] learn the dynamics among the multiple objects and track them at the same time. Multiple object tracking is also addressed along with human body pose estimation in [Gammeter et al. 2008]. Another example is that Choi and Savarese [2012] address MOT and action recognition in a unified framework.

REFERENCES

Saad Ali and Mubarak Shah. 2008. Floor fields for tracking in high density crowd scenes. In *ECCV*. 1–14.

Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. 2008. People-tracking-by-detection and people-detection-by-tracking. In *CVPR*. 1–8.

Anton Andriyenko and Konrad Schindler. 2011. Multi-target tracking by continuous energy minimization. In *CVPR*. 1265–1272.

Anton Andriyenko, Konrad Schindler, and Stefan Roth. 2012. Discrete-continuous optimization for multi-target tracking. In *CVPR*. 1926–1933.

Ben Benfold and Ian Reid. 2011. Stable multi-target tracking in real-time surveillance video. In *CVPR*.

Jerome Berclaz, Francois Fleuret, and Pascal Fua. 2006. Robust people tracking with global trajectory optimization. In *CVPR*. 744–750.

J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. 2011. Multiple object tracking using k-shortest paths optimization. *TPAMI* 33, 9 (2011), 1806–1819.

Margrit Betke, Diane E Hirsh, Angshuman Bagchi, Nickolay I Hristov, Nicholas C Makris, and Thomas H Kunz. 2007. Tracking large variable numbers of objects in clutter. In *CVPR*. 1–8.

Charles Bibby and Ian Reid. 2008. Robust real-time visual tracking using pixel-wise posteriors. In *ECCV*.

Biswajit Bose, Xiaogang Wang, and Eric Grimson. 2007. Multi-class object tracking algorithm that handles fragmentation and grouping. In *CVPR*. 1–8.

Michael D Breitenstein, Fabian Reichlin, Bastian Leibe, Esther Koller-Meier, and Luc Van Gool. 2009. Robust tracking-by-detection using a detector confidence particle filter. In *ICCV*. 1515–1522.

William Brendel, Mohamed Amer, and Sinisa Todorovic. 2011. Multiobject tracking as maximum weight independent set. In *CVPR*. 1273–1280.

G. Brostow and R. Cipolla. 2006. Unsupervised bayesian detection of independent motion in crowds. In *CVPR*.

A. Butt and R. Collins. 2013. Multi-target Tracking by Lagrangian Relaxation to Min-Cost Network Flow. In *CVPR*.

Joshua Candamo, Matthew Shreve, Dmitry B Goldgof, Deborah B Sapper, and Rangachar Kasturi. 2010. Understanding transit scenes: A survey on human behavior-recognition algorithms. *IEEE Transactions on Intelligent Transportation Systems* 11, 1 (2010), 206–224.

Kevin Cannons. 2008. A review of visual tracking. *Dept. Comput. Sci. Eng., York Univ., Toronto, Canada, Tech. Rep. CSE-2008-07* (2008).

Rich Caruana. 1997. Multitask learning. *Machine learning* 28, 1 (1997), 41–75.

Ting-Hsun Chang, Shaogang Gong, and Eng-Jon Ong. 2000. Tracking Multiple People Under Occlusion Using Multiple Cameras.. In *BMVC*. 1–10.

Wongun Choi and Silvio Savarese. 2010. Multiple target tracking in world coordinate with single, minimally calibrated camera. In *ECCV*. 553–567.

Wongun Choi and Silvio Savarese. 2012. A unified framework for multi-target tracking and collective activity recognition. In *ECCV*. 215–230.

N. Dalal and B. Triggs. 2005. Histograms of oriented gradients for human detection. In *CVPR*.

Caglayan Dicle, Mario Sznaier, and Octavia Camps. 2013. The Way They Move: Tracking Multiple Targets with Similar Appearance. (2013).

Andreas Ess, Bastian Leibe, Konrad Schindler, and Luc Van Gool. 2008. A mobile vision system for robust multi-person tracking. In *CVPR*. 1–8.

Andreas Ess, Bastian Leibe, Konrad Schindler, and Luc Van Gool. 2009. Robust multiperson tracking from a mobile platform. *TPAMI* (2009), 1831–1846.

Andreas Ess, Bastian Leibe, and Luc Van Gool. 2007. Depth and appearance for mobile scene analysis. In *ICCV*. 1–8.

Theodoros Evgeniou and Massimiliano Pontil. 2004. Regularized multi–task learning. In *ACM SIGKDD*.

P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. 2010. Object detection with discriminatively trained part-based models. *TPAMI* 32, 9 (2010), 1627–1645.

Pedro F Felzenszwalb and Daniel P Huttenlocher. 2006. Efficient belief propagation for early vision. *IJCV* (2006), 41–54.

Francois Fleuret, Jerome Berclaz, Richard Lengagne, and Pascal Fua. 2008. Multicamera people tracking with a probabilistic occupancy map. *TPAMI* (2008), 267–282.

David A Forsyth, Okan Arikan, Leslie Ikemoto, James O'Brien, Deva Ramanan, and others. 2006. Computational Studies of Human Motion: Part 1, Tracking and Motion Synthesis. *Foundations and Trends® in Computer Graphics and Vision* 1, 2-3 (2006), 77–254.

Thomas E Fortmann, Yaakov Bar-Shalom, and Molly Scheffe. 1983. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering* (1983), 173–184.

Katerina Fragkiadaki, Weiyu Zhang, Geng Zhang, and Jianbo Shi. 2012. Two-granularity tracking: Mediating trajectory and detection graphs for tracking under occlusions. In *ECCV*. 552–565.

Stephan Gammeter, Andreas Ess, Tobias Jäggli, Konrad Schindler, Bastian Leibe, and Luc Van Gool. 2008. Articulated multi-body tracking under egomotion. In *ECCV*. 816–830.

Dariu M Gavrila and Stefan Munder. 2007. Multi-cue pedestrian detection and tracking from a moving vehicle. *IJCV* (2007), 41–59.

Jan Giebel, Darin M Gavrila, and Christoph Schnörr. 2004. A bayesian framework for multi-cue 3d object tracking. In *ECCV*. 241–252.

Bohyung Han, Seong-Wook Joo, and Larry S Davis. 2007. Probabilistic fusion tracking using mixture kernel-based Bayesian filtering. In *ICCV*. 1–8.

Dirk Helbing and Peter Molnar. 1995. Social force model for pedestrian dynamics. *Physical review E* 51, 5 (1995), 4282.

Joao F Henriques, Rui Caseiro, and Jorge Batista. 2011. Globally optimal solution to multi-object tracking with merged measurements. In *ICCV*. 2470–2477.

Robin Hess and Alan Fern. 2009. Discriminatively trained particle filters for complex multi-object tracking. In *CVPR*. 240–247.

Martin Hofmann, Daniel Wolf, and Gerhard Rigoll. 2013. Hypergraphs for Joint Multi-View Reconstruction and Multi-Object Tracking. In *CVPR*.

Min Hu, Saad Ali, and Mubarak Shah. 2008. Detecting global motion patterns in complex videos. In *ICPR*.

W. Hu, X. Li, W. Luo, X. Zhang, S. Maybank, and Z. Zhang. 2012. Single and multiple object tracking using log-Euclidean Riemannian subspace and block-division appearance model. *TPAMI* (2012).

Weiming Hu, Tieniu Tan, Liang Wang, and Steve Maybank. 2004. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on SMC-C* (2004), 334–352.

Chang Huang, Bo Wu, and Ramakant Nevatia. 2008. Robust object tracking by hierarchical association of detection responses. In *ECCV*. 788–801.

Katsuhiko Ishiguro, Takeshi Yamada, and Naonori Ueda. 2008. Simultaneous clustering and tracking unknown number of objects. In *CVPR*. 1–8.

Hamid Izadinia, Imran Saleemi, Wenhui Li, and Mubarak Shah. 2012. (MP)2T: Multiple People Multiple Parts Tracker. In *ECCV*. 100–114.

Hao Jiang, Sidney Fels, and James J Little. 2007. A linear programming approach for multiple object tracking. In *CVPR*. 1–8.

Yonggang Jin and Farzin Mokhtarian. 2007. Variational particle filter for multi-object tracking. In *ICCV*.

Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. 2012. Tracking-learning-detection. *TPAMI* 34, 7 (2012), 1409–1422.

Rangachar Kasturi, Dmitry Goldgof, Padmanabhan Soundararajan, Vasant Manohar, John Garofolo, Rachel Bowers, Matthew Boonstra, Valentina Korzhova, and Jing Zhang. 2009. Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. *TPAMI* (2009), 319–336.

Amit Kumar KC and Christophe De Vleeschouwer. 2013. Discriminative Label Propagation for Multi-Object Tracking with Sporadic Appearance Features. In *ICCV*.

Bernardin Keni and Stiefelhagen Rainer. 2008. Evaluating multiple object tracking performance: the CLEAR MOT metrics. *EURASIP Journal on Image and Video Processing* (2008).

In Su Kim, Hong Seok Choi, Kwang Moo Yi, Jin Young Choi, and Seong G Kong. 2010. Intelligent visual surveillanceA survey. *International Journal of Control, Automation and Systems* 8, 5 (2010), 926–939.

Louis Kratz and Ko Nishino. 2010. Tracking with local spatio-temporal motion patterns in extremely crowded scenes. In *CVPR*. 693–700.

Louis Kratz and Ko Nishino. 2012. Tracking pedestrians using local spatio-temporal motion patterns in extremely crowded scenes. *TPAMI* (2012), 987–1002.

Cheng-Hao Kuo, Chang Huang, and Ram Nevatia. 2010. Multi-target tracking by on-line learned discriminative appearance models. In *CVPR*. 685–692.

Cheng-Hao Kuo and Ram Nevatia. 2011. How does person identity recognition help multi-person tracking?. In *CVPR*. 1217–1224.

Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. 2006. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, Vol. 2. 2169–2178.

Bastian Leibe, Konrad Schindler, Nico Cornelis, and Luc Van Gool. 2008. Coupled object detection and tracking from static cameras and moving vehicles. *TPAMI* 30, 10 (2008), 1683–1698.

Bastian Leibe, Konrad Schindler, and Luc Van Gool. 2007. Coupled detection and trajectory estimation for multi-object tracking. In *ICCV*. 1–8.

Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. 2007. Crowds by example. In *Computer Graphics Forum*. 655–664.

Xi Li, Weiming Hu, Chunhua Shen, Zhongfei Zhang, Anthony Dick, and Anton Van Den Hengel. 2013. A survey of appearance models in visual object tracking. *ACM Transactions on Intelligent Systems and Technology (TIST)* 4, 4 (2013), 58.

Yuan Li, Chang Huang, and Ram Nevatia. 2009. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *CVPR*. 2953–2960.

Ye Liu, Hui Li, and Yan Qiu Chen. 2012. Automatic tracking of a large number of moving targets in 3d. In *ECCV*. 730–742.

David G Lowe. 2004. Distinctive image features from scale-invariant keypoints. *IJCV* 60, 2 (2004), 91–110.

Wei-Lwun Lu, Jo-Anne Ting, James J Little, and Kevin P Murphy. 2013. Learning to Track and Identify Players from Broadcast Sports Videos. *TPAMI* 35, 7 (2013), 1704–1716.

Wenhan Luo and Tae-Kyun Kim. 2013. Generic Object Crowd Tracking by Multi-Task Learning. In *BMVC*.

Wenhan Luo, Tae-Kyun Kim, Bjorn Stenger, Xiaowei Zhao, and Roberto Cipolla. 2014. Bi-label Propagation for Generic Multiple Object Tracking. In *CVPR*.

A. Milan, S. Roth, and K. Schindler. 2014. Continuous Energy Minimization for Multitarget Tracking. *TPAMI* (2014), 58–72.

A. Milan, K. Schindler, and S. Roth. 2013. Detection- and Trajectory-level Exclusion in Multiple Object Tracking. In *CVPR*.

Dennis Mitzel, Esther Horbert, Andreas Ess, and Bastian Leibe. 2010. Multi-person tracking with sparse detection and continuous segmentation. In *ECCV*. 397–410.

Dennis Mitzel and Bastian Leibe. 2011. Real-time multi-person tracking with detector assisted structure propagation. In *ICCV Workshops*. 974–981.

Philippos Mordohai and Gérard Medioni. 2010. Dimensionality estimation, manifold learning and function approximation using tensor voting. *JMLR* 11 (2010), 411–450.

Peter Nillius, Josephine Sullivan, and Stefan Carlsson. 2006. Multi-target tracking-linking identities using bayesian network inference. In *CVPR*. 2187–2194.

Kenji Okuma, Ali Taleghani, Nando De Freitas, James J Little, and David G Lowe. 2004. A boosted particle filter: Multitarget detection and tracking. In *ECCV*. 28–39.

Youngmin Park, Vincent Lepetit, and Woontack Woo. 2008. Multiple 3D Object tracking for augmented reality. In *IEEE/ACM International Symposium on Mixed and Augmented Reality*. 117–120.

Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. 2009. You'll never walk alone: Modeling social behavior for multi-target tracking. In *ICCV*. 261–268.

Stefano Pellegrini, Andreas Ess, and Luc Van Gool. 2010. Improving data association by joint modeling of pedestrian trajectories and groupings. In *ECCV*. 452–465.

AG Amitha Perera, Chukka Srinivas, Anthony Hoogs, Glen Brooksby, and Wensheng Hu. 2006. Multi-object tracking through simultaneous long occlusions and split-merge conditions. In *CVPR*. 666–673.

Patrick Pérez, Carine Hue, Jaco Vermaak, and Michel Gangnet. 2002. Color-based probabilistic tracking. In *ECCV*. 661–675.

Hamed Pirsiavash, Deva Ramanan, and Charless C Fowlkes. 2011. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*. 1201–1208.

Fatih Porikli, Oncel Tuzel, and Peter Meer. 2006. Covariance tracking using model update based on lie algebra. In *CVPR*, Vol. 1. 728–735.

Zhen Qin and Christian R Shelton. 2012. Improving multi-target tracking via social grouping. In *CVPR*.

Donald B Reid. 1979. An algorithm for tracking multiple targets. *IEEE Trans. Automat. Control* (1979).

Vladimir Reilly, Haroon Idrees, and Mubarak Shah. 2010. Detection and tracking of large number of targets in wide area surveillance. In *ECCV*. 186–199.

Mikel Rodriguez, Saad Ali, and Takeo Kanade. 2009. Tracking in unstructured crowded scenes. In *ICCV*.

Mikel Rodriguez, Josef Sivic, Ivan Laptev, and J-Y Audibert. 2011. Data-driven crowd analysis in videos. In *ICCV*. 1235–1242.

Michael S Ryoo and Jake K Aggarwal. 2008. Observe-and-explain: A new approach for multiple hypotheses tracking of humans and objects. In *CVPR*. 1–8.

Paul Scovanner and Marshall F Tappen. 2009. Learning pedestrian dynamics from the real world. In *ICCV*.

Khurram Shafique, Mun Wai Lee, and Niels Haering. 2008. A rank constrained continuous formulation of multi-frame multi-target tracking problem. In *CVPR*. 1–8.

Jianbo Shi and Carlo Tomasi. 1994. Good features to track. In *CVPR*. 593–600.

Xinchu Shi, Haibin Ling, Junliang Xing, and Weiming Hu. 2013. Multi-target Tracking by Rank-1 Tensor Approximation. In *CVPR*.

Guang Shu, Afshin Dehghan, Omar Oreifej, Emily Hand, and Mubarak Shah. 2012. Part-based multiple-person tracking with partial occlusion handling. In *CVPR*. 1815–1821.

Guang Shu, Afshin Dehghan, and Mubarak Shah. 2013. Improving an Object Detector and Extracting Regions using Superpixels. In *CVPR*. 3721–3727.

Bi Song, Ting-Yueh Jeng, Elliot Staudt, and Amit K Roy-Chowdhury. 2010. A stochastic graph evolution framework for robust multi-target tracking. In *ECCV*. 605–619.

Daisuke Sugimura, Kris Makoto Kitani, Takahiro Okabe, Yoichi Sato, and Akihiro Sugimoto. 2009. Using individuality to track individuals: clustering individual trajectories in crowds using local appearance and frequency trait. In *ICCV*. 1467–1474.

Zehang Sun, George Bebis, and Ronald Miller. 2006. On-road vehicle detection: A review. *TPAMI* 28, 5 (2006), 694–711.

Valtteri Takala and Matti Pietikainen. 2007. Multi-object tracking using color, texture and motion. In *CVPR*.

Siyu Tang, Mykhaylo Andriluka, Anton Milan, Konrad Schindler, Stefan Roth, and Bernt Schiele. 2013. Learning People Detectors for Tracking in Crowded Scenes. In *ICCV*.

Siyu Tang, Mykhaylo Andriluka, and Bernt Schiele. 2014. Detection and Tracking of Occluded People. *IJCV* (2014).

Carlo Tomasi and Takeo Kanade. 1991. *Detection and tracking of point features*. School of Computer Science, Carnegie Mellon Univ.

Oncel Tuzel, Fatih Porikli, and Peter Meer. 2006. Region covariance: A fast descriptor for detection and classification. In *ECCV*. 589–600.

Stefan Walk, Nikodem Majer, Konrad Schindler, and Bernt Schiele. 2010. New features and insights for pedestrian detection. In *CVPR*. 1030–1037.

Xiaogang Wang. 2013. Intelligent multi-camera video surveillance: A review. *PRL* (2013), 3–19.

Zheng Wu, Thomas H Kunz, and Margrit Betke. 2011. Efficient track linking methods for track graphs using network-flow and set-cover techniques. In *CVPR*. 1185–1192.

Zheng Wu, Ashwin Thangali, Stan Sclaroff, and Margrit Betke. 2012. Coupling detection and data association for multiple object tracking. In *CVPR*. 1948–1955.

Junliang Xing, Haizhou Ai, and Shihong Lao. 2009. Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses. In *CVPR*. 1200–1207.

Junliang Xing, Haizhou Ai, Liwei Liu, and Shihong Lao. 2011. Multiple player tracking in sports video: A dual-mode two-way bayesian inference approach with progressive observation modeling. *TIP* 20, 6 (2011), 1652–1667.

Kota Yamaguchi, Alexander C Berg, Luis E Ortiz, and Tamara L Berg. 2011. Who are you with and Where are you going?. In *CVPR*. 1345–1352.

Bo Yang, Chang Huang, and Ram Nevatia. 2011. Learning affinities and dependencies for multi-target tracking using a crf model. In *CVPR*. 1233–1240.

Bo Yang and Ram Nevatia. 2012a. Multi-target tracking by online learning of non-linear motion patterns and robust appearance models. In *CVPR*. 1918–1925.

Bo Yang and Ram Nevatia. 2012b. An online learned CRF model for multi-target tracking. In *CVPR*.

Bo Yang and Ram Nevatia. 2012c. Online learned discriminative part-based appearance models for multi-human tracking. In *ECCV*. 484–498.

Changjiang Yang, Ramani Duraiswami, and Larry Davis. 2005. Fast multiple object tracking via a hierarchical particle filter. In *ICCV*. 212–219.

Ming Yang, Fengjun Lv, Wei Xu, and Yihong Gong. 2009. Detection driven adaptive multi-cue integration for multiple human tracking. In *ICCV*. 1554–1561.

Ming Yang, Ting Yu, and Ying Wu. 2007. Game-theoretic multiple target tracking. In *ICCV*. 1–8.

Alper Yilmaz, Omar Javed, and Mubarak Shah. 2006. Object tracking: A survey. *Acm computing surveys (CSUR)* 38, 4 (2006), 13.

Qian Yu, Gérard Medioni, and Isaac Cohen. 2007. Multiple target tracking using spatio-temporal markov chain monte carlo data association. In *CVPR*. 1–8.

Ting Yu, Ying Wu, Nils O Krahnstoever, and Peter H Tu. 2008. Distributed data association and filtering for multiple target tracking. In *CVPR*. 1–8.

Amir Roshan Zamir, Afshin Dehghan, and Mubarak Shah. 2012. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *ECCV*. 343–356.

Beibei Zhan, Dorothy N Monekosso, Paolo Remagnino, Sergio A Velastin, and Li-Qun Xu. 2008. Crowd analysis: a survey. *MVA* (2008), 345–357.

Li Zhang, Yuan Li, and Ramakant Nevatia. 2008. Global data association for multi-object tracking using network flows. In *CVPR*. 1–8.

Lu Zhang and Laurens van der Maaten. 2013a. Preserving Structure in Model-Free Tracking. *TPAMI* (2013).

Lu Zhang and Laurens van der Maaten. 2013b. Structure preserving object tracking. In *CVPR*. 1838–1845.

Li Zhang, Bo Wu, and Ramakant Nevatia. 2007. Detection and tracking of multiple humans with extensive pose articulation. In *ICCV*. 1–8.

Xuemei Zhao, Dian Gong, and Gérard Medioni. 2012. Tracking using motion patterns for very crowded scenes. In *ECCV*. 315–328.

Bolei Zhou, Xiaoou Tang, and Xiaogang Wang. 2012. Coherent filtering: detecting coherent motions from crowd clutters. In *ECCV*. 857–871.

Bolei Zhou, Xiaogang Wang, and Xiaoou Tang. 2011. Random field topic model for semantic region analysis in crowded scenes from tracklets. In *CVPR*. 3441–3448.

Bolei Zhou, Xiaogang Wang, and Xiaoou Tang. 2012. Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents. In *CVPR*. 2871–2878.