IEICE TRANS. INF. & SYST., VOL.E98–D, NO.7 JULY 2015

SURVEY PAPER

The State-of-the-Art in Handling Occlusions for Visual Object Tracking*

Kourosh MESHGI[†a)], ***Nonmember and*** **Shin ISHII**[†], ***Member***

SUMMARY This paper reports on the trending literature of occlusion handling in the task of online visual tracking. The discussion first explores visual tracking realm and pinpoints the necessity of dedicated attention to the occlusion problem. The findings suggest that although occlusion detection facilitated tracking impressively, it has been largely ignored. The literature further showed that the mainstream of the research is gathered around human tracking and crowd analysis. This is followed by a novel taxonomy of types of occlusion and challenges arising from it, during and after the emergence of an occlusion. The discussion then focuses on an investigation of the approaches to handle the occlusion in the frame-by-frame basis. Literature analysis reveals that researchers examined every aspect of a tracker design that is hypothesized as beneficial in the robust tracking under occlusion. State-of-the-art solutions identified in the literature involved various camera settings, simplifying assumptions, appearance and motion models, target state representations and observation models. The identified clusters are then analyzed and discussed, and their merits and demerits are explained. Finally, areas of potential for future research are presented.

key words: *online visual tracking, occlusion detection, occlusion handling, occlusion reasoning*

1. Introduction

Object tracking is increasingly demanded in various applications ranging from human-computer interfaces, to crowd analysis, video processing, surveillance, automation, and medical purposes. This task involves keeping track of the spatial and temporal changes of one or multiple target(s) in a video sequence. A number of surveys of visual object tracking have been published to cover the state-of-the-art tracking algorithms from various viewpoints.

Numerous trackers have been presented to tackle various challenges in visual tracking, such as illumination changes, non-rigid deformations, fast motion, and so on, focusing on one or a few of them. However, no ultimate solution to overcome all of these challenges has been found. Most importantly, many trackers have ignored occlusion or handled only partial occlusion of the target, while occlusion is known to be one of the most challenging aspects of visual tracking.

Occlusion happens when a portion (or the whole) of the target disappears from the observed scene due to obstruction of the camera's line-of-sight to the target. This phenomenon appears due to numerous reasons and is frequently seen in real world video sequences. Complex interactions between objects, large displacement, shadow casts and dense crowds are instances of the cases in which temporal and/or spatial occlusions may occur. There are many difficult problems that trackers should address to handle occlusion completely, such as appearance change during occlusion, persistent occlusions, re-emergence of occluded object, emergence of the initially-occluded object in the middle of the scene, temporal silhouette merging of multiple objects, split observation of single objects, and fragmented trajectory of target objects.

In contrast with the wealth of literature in visual tracking, the occlusion problem has received little attention. Gabriel et al. [1] formalized the occlusion problem in terms of the notion of "blobs" (i.e., a group of objects) and proposed two ways to approach the problem: merge-split approach and straight-through approach. Recently Lee et al. [2] revisited the occlusion problem in object tracking, yet there is no clear understanding of the general occlusion problem [3]. Besides, the literature offers many diverse strategies, to which a comprehensive and systematic inspection seems necessary.

Our contributions in this review paper are summarized into three: (i) a comprehensive review of state-of-the-art techniques for occlusion detection, reasoning, and handling; (ii) a formal definition of challenges in occlusion problems; and (iii) a new approach to attribute the occlusions based on extent (severity), duration and complexity.

In this review, offline tracking is excluded, and we focus on online tracking in which objects are tracked in a frame-by-frame basis. Following this introduction, in Sect. 2 the occlusion problem is defined and categorized, then challenges within it are described thoroughly. The ways in which occlusion is tackled in the literature are comprehensively studied in Sect. 3, followed by Sect. 4 that skims the procedure and material to evaluate tracker performances under occlusion. Following the substantial literature review, effective approaches to robust tracking under occlusion and future research directions are discussed in Sect. 5.

2. Occlusion Taxonomy

Occlusion happens when observation of the target object (or key attributes to identify the target object) is not available in order for the camera to keep tracking the spatial location of the target while the target is still present in the designated

Manuscript received August 8, 2014.
Manuscript revised December 8, 2014.
Manuscript publicized March 27, 2015.

[†]The authors are with the Graduate School of Informatics, Kyoto University, Kyoto-shi, 606–8501 Japan.

[*]This work was supported by the Platform for Dynamic Approaches to Living System from MEXT, Japan, and partly by JSPS KAKENHI, No. 24300114.

a) E-mail: meshgi-k@sys.i.kyoto-u.ac.jp

DOI: 10.1587/transinf.2014EDR0002

scene [2]. From the camera viewpoint, several objects may be present in a single line of sight. Different depth ordering of target and distractor objects lead to partial or complete viewing obstruction of the target object [3].

2.1 Occlusion Categories

Early studies of visual tracking either ignored the occlusion problem or claimed to handle just partial occlusion owing to their robust design. Recent studies classify the occlusion into three general categories [4]: (i) *Dynamic (inter-object) occlusion* is the outcome of overlapping with another object, which is closer to the camera; (ii) *Scene (background) occlusion* happens because of (still) objects inside the background model that are actually located closer to the camera; (iii) *Apparent occlusion* occurs because non-visible regions emerge due to pose and/or shape changes, silhouette motion, out-of-plane rotations, shadows, or self-occlusions. Another type of occlusion may arise in multi-camera setups. An object is considered occluded between a time when it leaves the filed-of-view of a camera and another time when it enters the field-of-view of another camera or return to the previous one [2]. This type of occlusion is called *Blindspot Occlusion*. This categorization is based on the occluder class, which is unable to distinguish the performance of a tracker in different occlusion scenarios.

By attributing occlusion, the ability of a tracker to handle different types of occlusion can be assessed. We propose three intuitive attributes to describe each occlusion: extent, duration, and complexity. *Extent* of an occlusion is defined over the key features of the target. In partial occlusion, some of the key features of the target are hidden from the camera, while a full occlusion is the case that object is entirely invisible to the camera while knowing that the target is still in the scene. *Duration* of the occlusion can be short or long: Temporal or short occlusions happen frequently in an urban scene where the duration of occlusions are short and limited; Persistent or long occlusions, on the other hand, usually require dedicated treatments to fully employ the dynamics of the target. This is especially troublesome for generative model-based approaches that employ multiple hypotheses to find the target after occlusion [5]. Here, a consistent bound between the definitions of long and short could be important. It is reasonable to set the border threshold as a portion of unoccluded frames in which the target was observed, as the tracker accumulates information about target (e.g., 8% of unoccluded frames in [5]). Considering the *Complexity* of the occlusion, if one of the key characteristics (e.g., appearance, orientation, motion direction, size, number of blobs and distance from camera) of the target changes drastically during occlusion, the occlusion is complex, otherwise it is a simple one. By combining these attributes, 8 different occlusion kinds are characterized (Fig. 1).

While easier occlusion cases (e.g., simple temporal partial occlusion) are handled by many recent trackers, more complicated ones (e.g., complex persistent full occlusion) are rarely handled (Fig. 2), which emphasizes the impor-

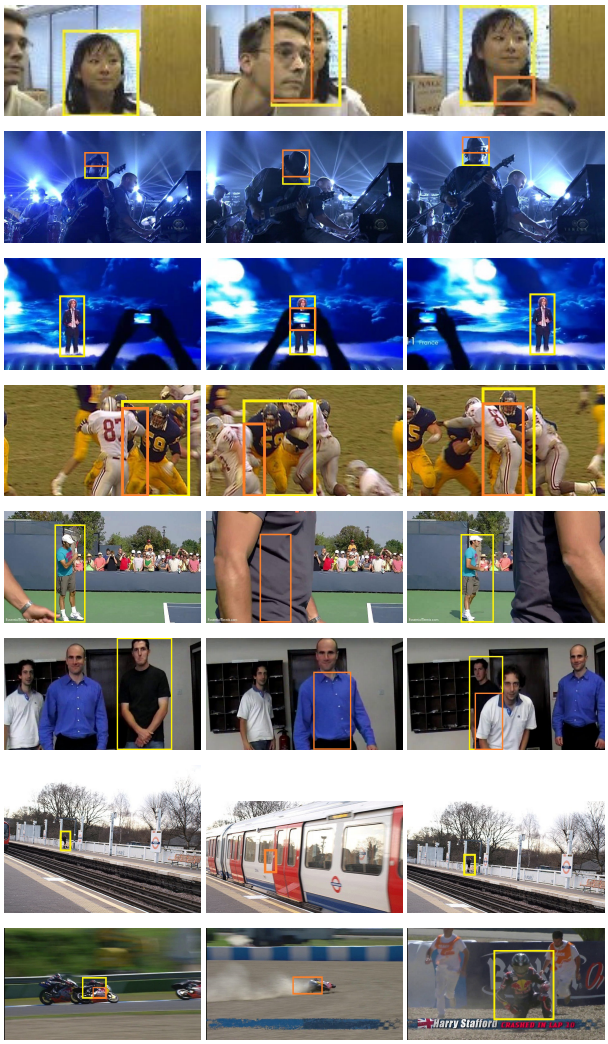


Fig. 1 Examples of eight occlusion cases yielded from three binary attributes: extent (**P**artial/**F**ull), duration (**T**emporal/**L**ong), and complexity (**S**imple/**C**omplex). Left and right panels illustrate before and after occlusion, respectively, while middle panel shows the scene during the occlusion. The yellow box indicates the target while orange parts show the occluded part of the target. – (row 1) PTS, (row 2) PTC, (row 3) PLS, (row 4) PLC, (row 5) FTS, (row 6) FTC, (row 7) FLS, (row 8) FLC.

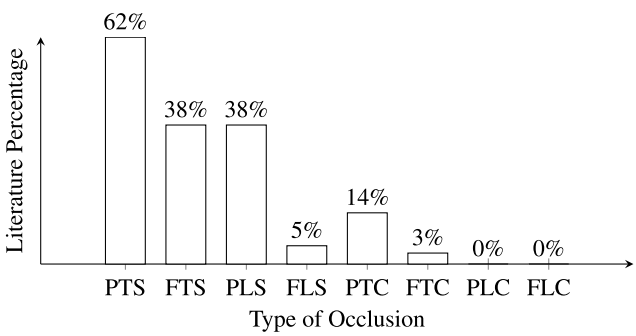


Fig. 2 Statistics of trackers in two recent benchmarks [6] and [7]. These two benchmarks examined 36 distinct trackers in total including the most recent and the well-established ones. Some of these total, including trackers were unable to continue tracking even in case of minor occlusions.

tance of redirecting more attention to tackle the latter ones.

2.2 Occlusion Induced Challenges

There are plenty of challenges that a tracker should deal with during and after an occlusion, some of which are less frequently attended in the literature.

During an occlusion, due to missing information, trackers' ability to localize their targets become limited. Some applications require the *localization of the occluded target*. The trackers are thus required to provide an estimation of the target location when it is partly or completely invisible.

The next problem is known as *split & merge problem*, which is essentially the task of classifying foreground pixels to the objects when the objects do not have one-to-one correspondence to foreground blobs, either due to that a single object appears as multiply fragmented foreground blobs (split) or when multiple objects form a single foreground blob (merge) [8]. This problem arises in the approaches where foreground pixels are separated from background ones and then target objects should be identified in the foreground blobs. The merge case happens when objects are grouped, are placed close to each other, interact, or overlap each other in the camera line-of-sight. On the other hand, partial occlusion and accidental alignment (e.g., portions of a moving object are accidentally very similar to those of objects in the background, resulting in misdetection of actually moving pixels) can fragment a silhouette into temporally or spatially separated elements [9]. To enumerate possible outcomes of split and merge phenomena in a scene, the notion of object support is presented. Object supports are hypothetical object regions, parts of which may not be visible due to occlusion. The study in [3] classified possible occlusion states into seven (OC-7) by considering the spatial relations between the object support and the detected foreground.

In addition to the object-foreground relationships, *varying number of objects* in the scene increases the complexity of the tracking scenario. A good multi-target tracker would be able to detect changing numbers of objects in the scene –by adding or removing objects when appropriate– and also able to handle both occlusion and split events. To handle the emergence of new objects, which is known as *birth problem*, trackers either use a global object detector, monitor possible object entry points [10], or initiate an object when an unidentified moving blob is detected. In the occlusion case, an object can appear in the middle of the scene as it separates from a group or emerges from an occlusion that has covered it since the beginning of the tracking session. Contrarily, an object may leave the scene, either for being hidden by an occluder or by joining a group, and hence becomes indistinguishable. This is called the *death problem*. To handle varying number of objects requires a concrete strategy to address birth and death problems.

When several targets overlap each other, it is sometimes necessary to determine their order along the camera's line-of-sight, i.e., in the depth direction. The task of disseminating the objects in the depth order is called *occlusion reasoning*. Having depth information in the case of stereo cameras, multiple cameras and range-finders makes the task trivial, but in the case of single fixed camera the problem turns to a big challenge.

Early trackers assumed that the target appearance does not change considerably throughout the tracking (e.g. [10]), while in real-world scenarios this assumption is typically violated. Hence, it is crucial to update the model to account for appearance variation. Updating the template with the latest observation is vulnerable to partial observations, in which the data is partly missing, or contaminated with irrelevant data from occluder or background. Especially in the case of partial occlusions, the trackers which only adopt model update without considering the observation reliability will fail, since their updated templates are apart from the current target appearance (i.e., drifted away). This argument holds for learning trackers in which non-target or occluded samples hinder model learning of the correct target appearance. To handle this *model drift problem*, the model should be updated slowly or selectively, to keep the memory of target appearance; contrarily a faster reactive update scheme is required to cope with frequent target variations [4].

After the complete resolution of the occlusion, in a phase called occlusion recovery, the tracker is expected to recover the target once again. The target may have changed during the occlusion or appear somewhere other than expected in the scene. A common problem in the occlusion recovery of multi-target trackers is *identity loss*. If the target ID is a new one, it means that the tracker suffers from re-identification problem, while giving wrong ID to the tracker (identity switch) could indicate model drift and/or confusion problem. The *re-identification problem* corresponds to trackers' strategy to handle birth & death and model drift problems. *Confusion problem* in multi-object trackers arises when the targets are "identical", in the sense that the same model is used to describe each of the targets [11]. In such cases, identity switch between them increases, so more discriminating feature or prior knowledge (such as estimated reappearance location) is required to resolve the problem. Furthermore, other problems such as unlikely trajectory for the similar targets [12], delayed recovery of target location and disturbed scale adaptation for the recovered objects [5] can be included to this list. Figure 3 illustrates the scope of main problems induced by occlusion in tracking timeline.

Next section provides a rich overview of research di-

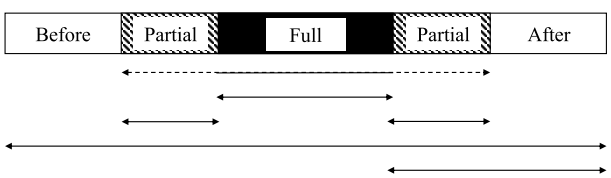


Fig. 3 The scope of occlusion induced challenges: (a) occluded target localization (b) varying number of objects (c) split & merge problem, occlusion reasoning (d) model drift problem (e) identity loss/switch, confusion problem.

rections to handle occlusion and related problems in visual tracking.

3. Handling Occlusions

Occlusions, if not handled, result in errors which may eventually cause the tracker to drift away from the target or lose it in the middle of tracking scenario. Some trackers assume controlled environments, in which several assumptions are satisfied. Such assumptions try to simplify the task, by introducing a new concept in tracking, or just by dealing with certain aspects of the occlusion. Instances of rather unrealistic assumptions are color-separability of objects [4], [13], smooth camera motion, stationary changing background, static target appearance, limited amount of spatial and temporal occlusion, limited interaction between objects, single class of objects in the scene, known birth/death time of objects or at least known number of them at each time [9]. There are other assumptions which facilitate tracking and hold in specific applications, such as stationary camera which eases background subtraction and sufficiently large objects to support strong appearance models. Moreover, there exist some assumptions derived from human visual system such as object permanence (i.e. partially or fully occluded objects, even though not observable, still exist in the close proximity of their occluders and move with them before they re-emerge) [13].

Occlusion handling is the task of minimizing the impact of occlusion on the tracking, which is achieved by granting robustness to the tracker against occlusion (passive handling) or preventing/compensating the disturbing effects of the occlusion (active handling). The occlusion handling has been more studied in discriminative tracking approaches, while generative models address the occlusion by definition since they maintain a large set of hypotheses, which by covering the target have a chance to survive occlusion and target recovery [4]. In this study, we present several directions identified in the literature to deal with occlusions.

3.1 Robust Representation

Holistic templates made from raw intensity values are the most popular representations for tracking. Many researchers, however, have pursued more advanced templates to handle different conditions of tracking, especially occlusions. To better account for appearance changes and handling occlusions, many ideas were applied to trackers (Table 1). Learning appearances introduced a new turn into appearance modeling as the researchers attempted to incorporate more intelligence into the trackers. A good representative of this category is discriminative models in which a binary classifier is trained online to discriminate the target from background [14]. Utilizing robust features and their combination is another trend in the literature. A good discussion on feature fusion in Bayesian framework was provided in [5]. Additionally, the fusion of different cues from a number of detection and tracking algorithms have been used to produce more robust trackers. There are plenty of robust representations against occlusions such as active contours, wavelet-filter based and covariance matrix representations [15]. Moreover, subspace-based tracking approaches [16] handled appearance changes well. To better handle appearance variations, some approaches recently proposed integrating multiple representations [17]. A good discussion over appearance models can be found in [18].

Table 1 Robust representations to handle occlusions.

Extension to Holistic Templates	Superpixels [26], [27]
	Local templates (patches) [28]
	Sparse representation [29], [30]
Learning Approaches	Integrate sample labeling into learning tracker [25]
	Codebook learning (using bag of features [31] or sparse representation [32])
	Feature selection [33]
	Discriminative classifiers (Learning from labeled and unlabeled data [34], incremental [14], [16], hough-forests [35])
Robust Features	Local features (Local templates, MSER, SIFT, SURF, corner feature [18])
	Saliency detection features [36]
	Combining features (Multi-cue raw-pixel, spatial-color histogram, appearance and depth domains features [5])
	Using different feature sets [37]
System-level Fusion	Self-organizing models and integration methods [38]
	Fusion of multiple detectors using RJ-MCMC [39]
	Hybrid system of trackers and detectors
	Sampling over a pool of simple trackers [40]

Model update problem can be efficiently handled in the appearance representation of the trackers using leaky memories [19], online mixture model [20], dictionary updating [21], online boosting [22], and incremental subspace update [16]. Discriminative models, on the other hand, strongly depend on the sample collection part to make the gradually trained classifier more robust [23]–[25]. Despite the progress in appearance models, preventing model drift in adaptive models is still far from perfect. Besides, insufficient attention has been paid to contingency methods such as model drift recovery, e.g., [26].

3.2 Motion Models

Estimating the state of the target has been proved useful in resolving issues of the occlusion especially in persistent or complex occlusions where the target is likely to continue changing while the model cannot be updated [6], [7]. Motion models enable trackers to fuse target motion with appearance model to continue predicting its state until it reappears.

If formulated in an *optimization* framework, the task is to find a plausible motion of the target which minimizes the distance of the path to the location of the target in consecutive frames while maintaining several constraints. If the objective function is differentiable with respect to motion parameters, gradient decent methods can locate the target efficiently [41], [42]. However, these objective functions

are usually non-convex or non-linear [7]. Unless an explicit occlusion term is embedded into the energy function, this class of motion models cannot handle full occlusions, e.g., in [43]. Still this category can handle partial complex occlusions in the cost of heavy computation.

Generally targets can be found close to its previous location, thus a *uniform search* around the location will find the target efficiently [25]. Such explicit position search may lose the targets whose motions are unexpectedly fast. To alleviate this problem, probabilistic *Gaussian motion models* are utilized [16], [29]. By putting more weight on locations closer to previous location, this solution poses more bias on the target motion than uniform search, hence, fails easier in the case of fast and abrupt motions [6]. Since the search area is fixed in these schemes, they are unable to handle persistent occlusions.

Dense sampling as utilized in [22], [24], [25] is one of the simplest solutions to handle large search space, but it suffers from high computational complexity. Hence, stochastic search algorithms have been widely used since they are relatively insensitive to local minima and computationally efficient. A natural choice of the sampling algorithm is the linear motion model, described by *Kalman filter*. Although prediction of motion as applied in [30] alleviated the fast motion problem, but is not applicable to general tracking easily. This is due to the fact that explicit motion modeling in complex scenes is difficult and not generalizable. Advanced Kalman filters can estimate trajectories in some cases of occlusion but degrade when the target objects, occlusion extents, or the distractors increase. Constant prediction of target location enables Kalman Filter motion models to handle full persistent and/or complex occlusions for motions with smooth and linear trajectory.

By handling non-linear non-Gaussian motions, *particle filters* surpass Kalman filters, handle temporal occlusions, and hence have become very popular. Particle filters are also widely used in multi-target tracking, being enhanced by partitioned sampling [11], Adaboost learning module to differentiate targets, ensemble of particle filters, and MCMC-based particle filters. These schemes are only capable of handling partial temporal occlusions. Nevertheless, if the correlations among objects are not exploited, generic importance sampling becomes inefficient as the number of targets increases, and in addition the joint state representation leads to high computational cost. Moreover, particle filter-based trackers with insufficient number of samples cannot generate statistically significant modes leading to fails in tracking multiple targets. Particle filters combine information obtained by sampling with assumed motion patterns (e.g., constant velocity) so they suit partial or temporal full occlusion scenarios. Yet, persistent or complex occlusions impede particle filter trackers.

Motion estimation is approached by *parametric motion models* that utilized predictors such as linear regression techniques or parametric second order linear system with online parameter tuning [44]. More rich models describe rotation [25], scale [41], shear and affine or 2D projective deformations of the target [45]. They are especially useful for complex partial and self occlusions. Motion prediction governed by *optical flow* is an interesting alternative [46]. Another attractive idea is to simultaneously perform tracking and detection as in [34]. This approach is proved to be very successful in handling various forms of partial occlusions, either persistent or complex.

A trending idea is to use *context information* to assist trackers, especially when the target is fully occluded or leaves the image region. Information such as local visual information surrounding the target [12], [47] or auxiliary objects in the scene [48] is recognized to be useful in this regard. Such knowledge empowers the tracker to deal with persistent full occlusions [7], and solves the invisible object localization problem mentioned earlier.

Motion estimation is not limited to the whole objects, as sub-object extensions can be found in the literature. Using a separate Kalman filter for every image patch [49] and tracking circular redundant image patches with mean-shift [3] are two good examples of such approaches. Complex temporal partial occlusion is effectively handled by those methods, while longer occlusions trouble trackers using this type of motion models. Although motion models estimate the target location and narrow down the whole scene to a smaller region of interest (ROI), errors in this prediction may result in losing the target permanently.

3.3 Foreground Tracking

Foreground trackers cast occlusion reasoning as a classical segmentation problem: classify foreground pixels into several sub-regions according to prior knowledge and track the targets based on segmented regions.

Motion-based trackers and appearance-based trackers are two families of such trackers. These methods track connected regions that roughly correspond to the 2D shape of objects based on their dynamic models [50]. The advantages of such models are their real-time applications, model-free description, and the large amount of information which is available to the tracker due to pixel level accuracy of the target description.

In order to detect foreground pixels, the background is usually subtracted from the video using methods like temporal median filter, Gaussian mixture models, and codebooks. To simplify the task, some studies assumed a stationary camera or color separability between objects. The foreground is then segmented into its primary entities, “blobs”. Each blob may contain more than two objects due to object proximity to one another, related occlusions and image noise, so that a blob may be composed of elements of one or more than two actual physical objects, which over time may shift from one observable blob to another [9]. An object, on the other hand, may be split into several blobs or merge together to form a bigger blob in various cases of occlusion (recall OC-7). Hence, the occlusion problem is reduced to an association problem of the blobs to the objects for which finding a unique solution in a real scene is challenging [51],

due to (i) image changes; (ii) the presence of non-rigid or articulated objects and their non-uniform features; (iii) multiple moving objects, especially similar or crossing objects; (iv) ambiguous matches, e.g., one blob corresponds to several objects, when objects split or merge; (v) erroneous segmentation; and (vi) changing features.

Solving the association problem formed the core concentration of many studies. Finding an exclusive correspondence between different objects by using joint probabilistic data association filter was one of many attempts to formulate a solution to this problem. Occlusion handling is considered in later studies. People interaction was typically handled using Bayesian networks. The split and merge problem, however, has been the biggest challenge of this category of trackers. Searching for all possible changes of blobs leads to an expensive combinatorial search [52]. Besides, due to fragmentation, target identities are switched by object interactions. Additionally, if the number of objects is varying or unknown, an ambiguity arises when using generic models to track objects that may be fragmented or grouped. This phenomenon is known as fragment-object-group ambiguity and demands an estimation of the number of objects and association of foreground blobs with objects simultaneously.

Early responses to the fragmentation problem were to avoid it using a distance-based criterion to cluster blobs and track those near each other [53], which leads to loss of resolution. This solution, however, is not effective in scenes where objects have grossly different sizes, because of unwanted grouping of objects especially in densely populated scenes [9]. Allowing merge/splits while tracking cluster of pixels gives rise to another ambiguity in which objects are not distinguished from fragments/groups, and/or object IDs before and after group interaction cannot be associated.

In order to distinguish a split blob from a single target or a single merged blob from several targets in multi target tracking, [54] used analytically solvable particle filter, [55] used thresholding over blob sizes, and [13] utilized a constrained optimization formulation exploiting object permanence constraint (explained earlier), [56] utilized a traditional Bayesian multi-target tracker over virtual blobs, and [57] formulated the problem as a multiple association problem. Methods based on pixel level regions were also proposed in the literature, e.g., [4]. As mentioned, a good foreground tracker should handle various number of targets. Handling the birth and death problem may assist this task. Listing active and passive objects [3] and growing and shrinking motion regions are two typical solutions for this problem.

3.4 Model-Based Tracking

Trackers of this class directly describe the target and attempt to track it explicitly throughout the scenario. In this case each blob contains only one object, and can be tracked individually without being merged in the possible event of occlusion. In order to describe the target, three methods dominate the literature: classification approaches, feature based techniques, and deformation models.

Trackers of the first category, known as tracking-by-detection approaches, contain a trained object detector or a generic object detector trained online during tracking. Such detectors are based on state-of-the-art machine learning techniques such as boosting [22], [58], semi-boosting [23], multi-instance boosting [24] and variations of support vector machines, e.g., [25], [59], [60].

Features which are almost invariant to the appearance change, pose or occlusion play a crucial role in robust tracking under occlusions. Haar-like features [22], histogram-based appearance features (e.g., HOG), histogram of relative optical flow (HOF) [61], and features learned from deep learning models, depth, segmentation and motion [62] are involved in such successful features. Features are either chosen manually or automatically from a set of features [63].

Deformable models have advantage when tracking non-rigid objects, by employing high resolution prior knowledge where all motions and appearances of the model components are well-defined and by regarding occlusions as missing information in the process of tracking.

Generic detectors model the whole object in a single template. Such detectors assume that objects are fully visible so their performance degrades in the case of partial occlusions [59]. On the other hand, part-based models which mainly model the translational deformation of parts are typically more resilient against occlusions. Part-based deformable models such as [60] sum up the scores of part detectors, and the existence of the object in the input window is indicated with a relatively high total score. In this model, an occluded part may have very low score which ends up to a low total score. Therefore, some trackers rely on the detection score of the part to estimate its visibility [62], combining the responses of part detectors to form a joint likelihood model, or calculating a weight for part based on their appearance difference from background. In addition to known object parts, meaningless patches (fragments) are also tracked in [64] while independently moving entities are clustered probabilistically in an unsupervised fashion. The authors of [62] hypothesized that the key to successful detection of partially occluded humans is to utilize additional information about which body parts are occluded. Having such extra knowledge along with other information from motions, depth, and segmentation enables the tracker to compensate partial occlusion effectively; i.e., when the occluded parts are identified, their effect should be appropriately removed from the final combined score. Deformation score in combination with appearance score can also promote a more accurate tracking using part-based detectors [59], [60]. So far all studies shown above assumed independence among different object parts and hard-thresholded the detector score to determine visibility. A step toward more realistic assumptions was taken in [65] where an expert described the relationship between parts in a rule-based fashion, whose idea was later extended in [66] where the visibility relationships among parts were learnt systematically from training data using a discriminating deep model.

The recovery from occlusion in object trackers is straight forward. If the search area and target template perform well, the target is recovered immediately after it reappears. However, model drift reduces the discrimination power of the model so that more accurate search mechanism is required to compensate this artifact. Employments of context information, motion models, and auxiliary trackers are beneficial for empowering the tracker's search mechanism. To handle the model drift problem, forgetting memories for templates and sampling from non-occluded target appearance for learning module are two established solutions while advanced attempts to bring more robustness to template update have been made, e.g., [20].

3.5 Mode Switching Trackers

Every tracker has its own characteristic which suits for specific application and yet no dominant general purpose tracker is released by the community. Naturally, changing the circumstances may hinder tracking of some algorithms, while some others may work perfectly. For instance, a tracker may be good in handling variations in illumination, but may not necessarily be able to cope with appearance changes of the object caused by variations in object viewpoints. Also a tracker might predict motion to better anticipate its speed, but it may have difficulty in following bouncing objects. As an another example a tracker may make a detailed assumption of the appearance, but then may fail on an articulated object [6]. Balancing the trade-off between different trackers heavily depends on the task in hand and hence scenario conditions. One of the most disrupting changes in the environment is occlusion. Some researchers believe that by switching trackers, or adapting some tracker modules to new circumstances, better trackers can be constructed.

Switching between trackers have been studied in [50] where the tracker resorts to a particle filter tracker when the main part-based tracker undergoes some occlusion. In a study by Wu and Nevatia [67], a global part-based tracker switched to an individual mean-shift tracker for each part when the data association failed. Utilizing a mean-shift tracker in normal condition and switching to a particle filter in the case of poor performance are another successful solution [68]. Kwon and Lee [40] presented a switcher which chose only the required trackers, suitable for current condition, of tracking from a tracker pool.

Within trackers there are many modules which by operating differently cope better with the situation. Switching motion models and soft/hard switching between two sampling methods in a particle filter tracker as in [44] are good examples of such methods. These method are based on fixed switching criteria, which render them sensitive to parameter settings. To address this issue, [5] proposed an adaptive switching method which alters tracker dynamics, switches motion models and sampling methods, and recovers quickly from occlusion.

3.6 Multiple Camera Scenarios

Visual tracking with multiple cameras significantly reduces the challenges introduced by occlusion in different scenarios at the expense of simplicity of the tracking algorithm. In order to minimize the occlusion, cameras with face downward or omni-view (360°), ultra-wide bird-view cameras, multiple static cameras, stereo cameras, multiple automatically driven cameras, and even non-overlapping cameras have been used, each bringing its benefits and disadvantages into tracking [1], [2], [69].

With several cameras covering a scene from different viewpoints, a target that is invisible to one of the cameras may still be visible from other cameras, which reduces the probability of full occlusion. This observation also suggests that in multiview monitoring, the videos obtained from different cameras must be "fused" to handle occlusion [1]. Reasoning about occlusion relations between objects in such scenarios has been incorporated in several trackers using Bayesian networks.

3.7 Occlusion Detection

Despite its importance, occlusion detection is rarely addressed explicitly in the literature. It is understandable since the wide variety of occlusion scenarios makes it difficult to find a reliable occlusion detection metric. On top of that, not all of the occlusion detection methods are generic, and some of them is limited to specific application or tracker architecture (e.g., particle filter tracker).

In the case of foreground trackers, the ratio between the number of observable points (the one in the foreground blob) and points of the appearance model provides a clue about occlusion, as the low value of this ratio indicates the occlusion [4], [52]. Large deviations from appearance model [20], heuristic criteria on proximity and size changes of blobs, robust region description near keypoints, and thresholding the sum of likelihoods in particle filters [70] are good methods of detecting occlusions. More sophisticated method such as using structured sparse learning [21] or learning occlusions by likelihood [71] are modern solutions which are promising. Using histogram of depths to detect occlusions [72] is yet another recent approach which emerged after the advent of cheap range sensors.

Occlusion detection enables tracker to prepare for eminent occlusions, to change state during occlusion, and to monitor the target reappearance for a quick recovery. These favorable characters would resolve template update problem [19]: the tracker stops model update during full occlusion, or perform a partial model update if it can detect the occluded areas of target in the case of partial occlusion. Such knowledge is also crucial for handling varying number of objects and birth/death problem. Effectively managing the identities of objects throughout the course of occlusion (to prevent ID loss/switch) is another benefit of such detectors. Besides, such detector facilitate target recovery

right after the target reappears. For instance if the detector is global, it finds the target after the occlusion using a global search. Further application of occlusion detector is to signal motion models to prepare for occlusion state. For instance in [5], the ROI introduced by motion model is expanded gradually, once the occlusion is detected, in order to cover trajectory changes during complex persistent occlusions. Depending on actual applications, the occlusion pattern could be learnt from annotated data which will increase the robustness of this scheme. Monitoring other moving objects of the scene to predict possible occlusions, is another bright idea which elevates the effectiveness of occlusion detection.

3.8 Occlusion Reasoning

Occlusion reasoning is the process to determine the occlusion relationships between objects explicitly and then to localize the objects accurately. This is one of the most challenging problems in visual tracking because of partial visibility of occluded objects and ambiguous correspondence between objects and their features. Simplified versions of this task are handled for rigid objects using bounding contour of the motion mask, using 3D camera calibration information and using multiple calibrated cameras [2].

Pixel level analysis provides many insights into occlusion reasoning. Using probability maps [4], using "disputed" pixels [55], using non-linear feature voting strategy [51], assuming occlusion relationships as a function of only relations in previous state [73], or oppositely assuming it to be only dependent on the current state [74] are among successful solutions in handling occlusion reasoning.

Another approach to occlusion reasoning is to utilize layer representation. Layer information is crucial for estimating where a fully occluded object resides after its region splits [13]. Automatically decomposing the video into constituent layers sorted by depth, predicting self-occlusions using layered template and kinematic model, decomposing video into layers and employing EM to infer objects' appearances and motions, defining background layers, and ground plane constraints are instances of this genre [1], [13].

Inferring the occlusion relation of the targets is another popular approach found in the literature. Using Bayesian networks, competitive optimization using species based particle swarm optimization, and competitive game-theory based inference [75] indicate the large potential of using different AI solutions in handling this problem.

3.9 Summary of Literature

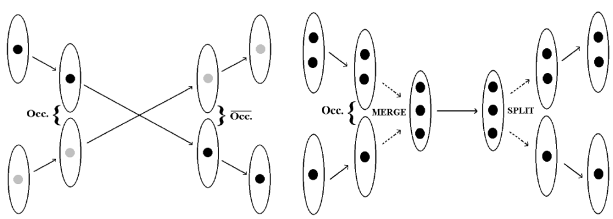
The diverse range of solutions proposed to tackle occlusion was characterized into eight major groups, each of which covers a different aspect of the occlusion problem. *Robust representations* provide robustness to partial observation along with other invariances such as rotation and illumination invariance, and are required for real-world trackers. Partial occlusions are handled by many modern representations, while complex occlusions still trouble many of such encodings [18]. Any problems in representation will result in model drift under partial or full occlusion and hence lead to track loss.

Motion models perform the smart selection of region of interest (ROI), in which target is expected to appear and shrink the search space significantly. This is specially useful in the case of complex partial occlusions, or even simple full occlusions. However, not all of motion models are eligible to handle persistent occlusions in which the object keeps moving while being invisible to the camera. Still hardly any of them are capable of dealing with complex persistent scenarios in which the target alters their course and speed while being invisible to the camera. Inappropriate motion model will result in track loss of the target, which requires additional mechanisms to relocate the target and continue tracking it.

Foreground tracking employs spatio-temporal segmentation techniques to embody the moving target and its accompanying objects. Using the extra "context" information in the blob facilitates handling full occlusions [47] while the fragmentation and merging events in the blob, often caused by complex partial occlusions, paralize this kind of scheme and thus require sophisticated designs. On the other hand varying number of objects complicates the assignment process in case of multi-target tracking. Yet this scheme provides a strong basis to handle simple partial occlusions and object interactions (Fig. 4 (b)).

Rapid expansion of the object detectors, especially with the advent of deep learning in computer vision [76], emphasizes on the role of *model-based tracking* that tries to find a match for the modeled target in the ROI. As the essence of tracking is locating the target in consecutive frames, this module plays a crucial part in most of the trackers, yet it is very vulnerable to occlusions. This scheme is defenseless against full occlusion, but is a powerful tool against partial occlusions, even the complex ones. Typically, more abstract models are robust against partial occlusion in the expense of higher mismatches. If the model needs to be updated frequently, it is subject to model drift problem which requires other mechanism to incorporate (Fig. 4 (a)).

When a strategy fails in tracking, *switching* to another strategy might help maintaining the performance of tracker. Complex partial occlusions, persistent occlusion, and even



(a) Model-based Tracking (b) Foreground Tracking

Fig. 4 Model-based versus foreground tracking. In the former each blob contains only one object while in the latter a blob involves parts of one or more objects [1].

full occlusion are some of cases where ordinary trackers rarely work and thus need special treatments. Switching mechanisms monitor switching criteria and switch between strategies. For instance in [44] trackers switch to more occlusion-robust strategy when dealing with full occlusion or in [5] the primary tracker is able to deal with partial occlusions, yet switches to another strategy when dealing with complex and/or persistent occlusions.

Occlusion detection enables trackers to take appropriate action, e.g., stop model update or start a global search for the object. There exist few attempts on full occlusion detectors while partial occlusion detection relies on detailed models which in turn require expensive calculation per frame and are not feasible for online tracking or complex occlusions.

Occlusion reasoning additionally brings about the depth order of the objects, which can be used for more intelligent tracking. Yet this category of schemes are working with simple partial and full occlusions and their performance still heavily depends on the scenario [2].

Finally, using *multiple cameras*, by increasing redundancy, shrinks the possibility of both full and partial occlusions. Table 2 summarizes the occlusion handling components and illustrates their relation to the occlusion attributes. This section demonstrates that handling occlusions requires multiple components to cooperate. For instance to handle complex and persistent occlusions, [5] combined robust representation, non-linear motion model, adaptive switching method, and occlusion detection. Moreover active handling of occlusion is achieved by schemes which actively change tracking strategy according to tracking scenario and input data. The prominent active occlusion handling schemes include model updating, adaptive motion modeling, adaptive strategy switching, occlusion detection, and occlusion reasoning.

The enormous amount of ideas into occlusion handling, reasoning and detection have made it clear that these tracking issues should gain more attention. To evaluate these ideas, however, an agreed evaluation framework is required to compare the efficiency and effectiveness of them.

Table 2 Occlusion handling components: extent (**P**artial/**F**ull), duration (**T**emporal/**P**ersistent), complexity (**S**imple/**C**omplex), and Active handling (**A**ctive/**P**assive).

Component	Extent	Duration	Complexity	Active
Robust Representation	P	-	S / C	P
Motion Models	F / P	T	S / C	P / A [1]
Foreground Tracking	P	T	S / C	P
Model-based Tracking	F / P	T / P [2]	S	P / A [2]
Mode Switching	F	-	S / C [3]	P / A [3]
Multiple Cameras	F / P	-	C	P [4]
Occlusion Detection	F / P	-	S / C [5]	A
Occlusion Reasoning	P	T	S / C	A

[1] *adaptive motion models*
[2] *with model update*
[3] *adaptive switching criteria*
[4] *here, fixed camera case is assumed.*
[5] *yet to be proposed*

4. Evaluation

Given the variety of occlusion circumstances in tracking, and the diverse solutions proposed to tackle this problem, it is surprising that the number of evaluation video sequences for this specific task is small. Moreover, a comprehensive and established standard to compare the ideas is lacking in the literature. This section gathers the attempts to promote occlusion handling by providing infrastructures (i.e., data and protocols) to evaluate the ideas comprehensively.

4.1 Criteria

Many measures for evaluating the tracking performance have been proposed, typically by comparison with ground truth considering the target presence [6]. This condition, however, reduces the applicability of those measures in the scenarios where the performance of the tracker under occlusion is also important. Ideally, a tracker is expected to track the target when it is present and to change its status to occlusion when the target is fully occluded. A good indicator of tracker's success in each frame is partial overlap that is defined as the ratio of spatial intersection between ground truth and system output over the spatial union of them. Defined in PASCAL framework, the overlap above 50% (overlap threshold = 0.5) is accepted as a tracking success which is later extended to account for occlusion handling [72]. VACE framework further extended it to multi-target tracking cases (*SFDA* metric [77]) and a threshold-free measure [7]. To account for important role-players of multi-target tracker performance such as number of objects detected and tracked, missed objects, false positives, fragmentation in both spatial and temporal dimensions, and localization error of detected objects in a single score, there is VACE *ATA* metric, which is an advanced version of the CLEAR metric with consistent object IDs [77]. Another metric is proposed in [28], which is the introduction of F-score into the tracking realm. An area based version of this score, $F1$-score, is later introduced by the same author. Good reviews over such criteria can be found in [6].

Regarding the output state of the tracker and the occlusion state of the ground truth, different kind of error can be imagined. Type I error occurs when the target is visible, but the tracker's output is far away from the target. Type II error occurs when the target is invisible, but tracker outputs a bounding box. Type III error occurs when the target is visible, but the tracker fails to give any output [72].

In order to demonstrate a way to analyze and measure occlusion robustness, here we formalize the occlusion problem and introduce a criterion to measure tracker's performance dealing with occlusions. Inspired by [72] and [77] here we propose a metric which supports multi-target tracking under occlusions. The metric accounts for tracker's accuracy and supports occlusions.

In a scenario with T frames, a total number of N targets are annotated which are not always present in the scene, either they enter/exit the scene in the middle of the scenario or they are occluded, thus a subset of those targets are visible in the frame $t \in \{1, \ldots, T\}$. The annotation of target $j \in \{1, \ldots, N\}$ in frame t is denoted by B^*_{jt} and its presence is denoted by the binary flag $s^*_{jt} \in 0, 1$, where one means the target is visible (at least partly) and zero otherwise. In the frame t, the tracker locates the target in area $\hat{B}_{jt}$ and $\hat{s}_{jt} = 1$ or announces that the target is not found/occluded by setting $\hat{s}_{jt} = 0$. The overlap of tracker's belief about the target extent and the ground truth is a good indicator of the tracker accuracy and calculated as $v_{jt} = |B^*_{jt} \cap \hat{B}_{jt}| \div |B^*_{jt} \cup \hat{B}_{jt}|$. Here, $\cap$ and $\cup$ are intersection and union operators for areas of the image, and $|.|$ operator counts the number of pixels embodied in an area. The score is comprised of four components: For each tracker at time t the score is negative when the visibility state of the target and tracker mismatches for a target (type II and III errors). This score is positive when the tracker's output and the annotation overlap or when the absence of a target in the scene is correctly detected. In order to prevent type I error, the overlap value is thresholded with τ using the Heaviside step function $\chi(.)$ The final score is the sum of scores for all targets averaged along the scenario.

$$R = \frac{1}{T} \sum_{t=1}^{T} \sum_{j=1}^{N} \Big(\hat{s}_{jt} s^*_{jt} \chi(v_{jt} - \tau) + (1 - \hat{s}_{jt})(1 - s^*_{jt}) - (1 - \hat{s}_{jt}) s^*_{jt} - \hat{s}_{jt}(1 - s^*_{jt}) \Big)$$

This criterion handles accuracy of tracking as well as changes in the number of objects detected and tracked, full occlusions, birth/death cases, ID loss/switch events, re-identification problem, confusion problem, and false occlusion alerts. It is an extension of the criterion introduced in [72] to handle multi-target tracking and punishes errors, and is more reliable than VACE ATA [77] because of handling type II and type III errors. Since some trackers may produce outputs that have small overlap ratio over all frames while others give large overlap on some frames and fail completely on the rest, τ must be treated as a variable to produce a fair comparison [72]. Accordingly, Wu et al. [7] proposed the success plot (success score versus threshold) and considered its area under curve (AUC) as the metric for measuring the performance. Using the proposed score R, the AUC of this plot is calculated as $AUC = \int_0^1 R(\tau) d\tau$.

4.2 Datasets

With the ever increasing volume of datasets released online, the lack of datasets to comprehensively evaluate occlusion becomes more evident. Many datasets for different computer vision task have been released so far, while a few of them referenced a portion of their videos as related to occlusion. Table 3 shows the list of datasets including videos with occlusion. It is clear that such datasets are not specifically focused on occlusion, with few videos including only a subset of possible occlusions. So far, valuable attempts have been made to alleviate this shortcomings such as [78] which created 64 simulation video sequences to experiment the effectiveness of each tracking method in various occlusion scenarios. The authors of this study tried to isolate other tracking challenges such as shadow, illumination changes and moving background from the occlusion scenarios. Also [72] provided an RGBD dataset with high diversity, including deformable objects, various occlusion conditions, and moving camera in different scenes.

Table 3 Datasets including videos for evaluating occlusion robustness of trackers.

Name	Description
PETS	Various tasks, 2001-9
ViSOR	Video surveillance, 2008
i-LIDS	Multiple camera tracking scenario, 2008
Caltech Pedestrian	Pedestrians, 2009
TUD-Brussels People [79]	Moving platform people detection, 2008
ALOV++	Challenging videos benchmark, 2010
TRECVid	Large video benchmark, 2010
RGDB People	Multi-Kinect videos of people, 2011
SimOcc [78]	64 Simulated occlusion videos, 2012
TB-50 [7]	Popular videos benchmark, 2013
Princeton RGBD [72]	Kinect video with occlusions, 2013

4.3 Guidelines from Benchmarks

Performance of the tracker varies in different scenarios and a fair comparison of them seems necessary to effectively evaluate trackers. To this end, several parallel benchmarks have been conducted in recent years. Wu et al. [7] carried out large scale experiments to understand how trackers work; especially they analyzed the initialization problem of object tracking comprehensively. A novel quantitative performance evaluation methodology was proposed in [80], which considered the tracking accuracy and durability to compare adaptive trackers versus non-adaptive ones. The experimental survey presented in [6] aimed to evaluate trackers systematically and experimentally on large number of video fragments as they believed that most of the studies used less than ten videos or special datasets for their evaluation.

The performance anaylysis on the occlusion subset of videos in [7] revealed that the trackers detailed in [25], [34], [81]–[83] outperformed others. The results reinforce the role of structured learning and sparse representation in occlusion handling. It was also deduced that local sparse representations are more effective than the ones with holistic sparse templates (e.g., [84]). The results further revealed that trackers tend to perform better in short sequences rather than long scenarios and the background information is critical for effective tracking. The data showed that motion model or dynamic model is crucial for object tracking, especially when the target motion is large or abrupt.

For modern trackers such as [25], [34], [40], [85], [86], under no/little motion relative to camera, even full temporal occlusions are not much of problem according to [6]. This benchmark demonstrated that occlusion with less than

30% of target extent is now considered a solved problem. In contrast, it revealed that for videos with large motion and full occlusion, most trackers have difficulty in reacquiring the target when it reappears. The results proved that [34] is overally the best tracker to handle occlusions, yet there is no tracker to handle all occlusion scenarios perfectly. Moreover, this study suggested that [35] works well for some occlusion scenarios characterized by small and fast-moving targets.

The focus of [80] is to compare the adaptive trackers with non-adaptive ones in several scenarios including partial occlusions and temporal full occlusions. For the partial occlusions [25] exhibits the best performance followed by [16] whereas other trackers [24], [64], [87] are effective only in some partial occlusion scenarios. Maintaining a good balance between stability and adaptation to appearance changes and a stable model update strategy to compensate for the over-simplistic state sampling strategy are recognized as the key elements of a successful tracker handling occlusions. Some other partly successful strategies in handling partial occlusions are identified as online feature selection [24] and deploying external labelers for the sampling and labeling stages [23], [88]. The authors argued that strong priors on target appearance are effective solutions for partial occlusions, but limit adaptability to appearance changes. They also mentioned that [34] is effective at handling occlusions but is unable to handle permanent appearance changes. Besides, this study emphasizes the role of stable model update strategy in occlusion handling (e.g., subspace or manifold update when using target-wise features in [16]) and hold overfitting to appearance changes responsible for model drift in trackers like [17]. To handle this overfitting issue, the study suggests the temporal smoothness to be enforced on the model update. Surprisingly, the study showed that a non-adaptive solution like [64] is more effective than many adaptive trackers (e.g., [16], [24]), during partial occlusions. For the temporal full occlusions the results suggest that current scale-adaptive trackers (except [34]) cannot handle this kind of occlusion well. During this experiment, even the temporal occlusion is too long for [16] to keep up, [16], [24], [34] have problems in dealing with rapid motion and consequent motion blur, and [25] has problems in handling out-of-plane rotations. Based on experiment outcomes, this study votes for [25] because of its effectiveness in dealing with rare and continuous appearance changes and robustness to partial and total occlusions and misaligned initial states.

The brief benchmark in [72] brings the flavor of error-type analysis into benchmarks. High Type II errors (where the target is occluded, but the tracker outputs a bounding box) is typical for the trackers without an explicit occlusion handling mechanism like [24], [25], [45], [89]. High Type III errors (where the object is present but the tracker cannot locate it) suggest that trackers like [23], [34] are sensitive to target appearance change or partial occlusion. Conservative approaches, which do not produce output with low confidence, often fail in tracking the target and fall into this type of error.

According to these benchmarks, TLD [34], Struck [25], VTS [40], ColorFBT [85], L1O [86], SCM [81], LSK [82], ASLA [83] and FBT [14] are successful trackers when dealing with occlusions, with the first two being emphasized for their robust performance even in challenging scenarios. Moreover, mix results have been reported about IVT [16] and FragTrack [64] which require further investigations[†].

4.4 Occlusion Scenarios

There are numerous possibilities for occlusion scenarios, each of which has different characteristics and requires a special kind of treatment. Several attempts have been made to describe the occlusion space, but only scratched its surface. Lee et al. [78] simulates 64 occlusion scenarios using different motion patterns (8 uniform trajectory and 8 non-uniform trajectory) and occlusion types (no occlusion, partial occlusion, full occlusion, long occlusion). In their experiment they used two rigid convex objects so that many complexities of dealing with non-rigid objects are relaxed. Meanwhile they tried to keep other factors in the scene constant. These scenarios were exaggeratedly simplified, contained no complex occlusion and even the duration of long occlusion was not enough to make the mean-shift tracker [42] drift away from the target - which is the famous shortcoming of this tracker against full occlusions.

In another study, Guha et al. [90] claimed that all occlusions can only have 14 states (OCS-14) regarding to the target being static/dynamic, degree of visibility, and state of object isolation. However, since different scenarios may result in similar states, the tracker may require different modules to be embedded to handle the same type of occlusion. The same argument applies to 3 attributes introduced for occlusion in this article. These attributes (extent, duration, and complexity) yield 8 states that only describe the occlusions, but give no information about the way of occlusions.

To approach the problem of enumerating all possible occlusion scenarios, it is important to analyze the role-players involved in scene formation: camera, light, object, and scene background [91].

Cameras observe the scene and provide the essential information to the tracker. The data obtained from single camera, lacks geometric information since it maps 3D world onto a 2D image plane. Yet the viewing angle of the camera significantly affects the scene complexity, ranging from overhead cameras in surveillance scenarios, to low altitude cameras such as those mounted on mobile robots. Stereo vision and RGB-D sensors try to compensate this shortcoming by providing partial 3D information about the scene, but provide tracker with the valuable depth information for visible surfaces. Multiple camera configurations, especially overlapping ones, shrink the chance of occlusion, but require pre-calibration or real-time image registration among

[†]Here, the most common tracker names in the literature or the naming in corresponding benchmark is used.

cameras. Camera movements complicate the tracking task drastically as they introduce rapid pose change, dynamic background, and different levels of self- and scene- occlusions. If there are more than one camera in the scenario, the depth separability becomes an important factor in creating novel scenarios. Occlusions due to objects which are spatially far but overlapped in image plane (i.e., have large distance in z direction) are more easy to handle for occlusion reasoning methods.

Illumination of the scene is another important role-player in the tracker performance. Sudden changes in illumination degrade the performance of both model-based and foreground tracking modules. Cast shadows, by altering the object appearance, imposing self-occlusion, and causing shadow-to-object resemblance problem, further challenge appearance trackers as well as introducing a complex non-linearity to motion models. Outdoor scenes and indoor stage plays are two typical scenarios subject to drastic light change which involves full occlusions or partial occlusions due to cast shadows.

Scenes are another source of frustration for trackers, since they are composed of background and non-target objects which may provide scene-to-object occlusions. Urban scenes especially are full of distractions, clutter, and partial to full occlusions caused by static background objects (e.g. traffic signs, benches, etc.) and non-target ones (e.g., cars on the street).

The targets themselves are a great source of variation. Number of targets in a scenario can be one or more, and a varying number of targets in a scene (e.g., surveillance scenarios) challenges tracker, especially when the scenario embodies occlusions. Detection/tracking in crowded scenes (e.g., airports, subway platforms, etc.) are now getting more attention in literature. Numerous instances of partial to full occlusions in a crowded scene require dedicated treatments which introduce concepts like crowd modeling, collective action recognition, interaction analysis, etc. to assist occlusion handling. A survey on crowd tracking readers can be seen on [92]. The variety of object classes [9] (e.g., in point-of-view video footage from urban scenes), and confusion problem (e.g., tracking a student in uniform in a school, a biker in Tour de France) are two more common challenges happening in real-world scenarios.

Targets which undergo non-rigid transformations and out-of-plane and those taking complicated poses and articulated motions create self-occlusions, which trouble model-based tracking. This is a common challenge in tracking non-rigid targets such as faces and pedestrians. Physical model shortcomings and high computational cost for heavily detailed models are common in such cases.

Although motion models try to capture the dynamics of the target, relative motion of the target to the camera still challenges many trackers. Bouncing, shaking, and other kinds of sudden trajectory changes along with partial or full occlusions are considered as the most challenging scenarios in many studies. Extreme cases of such scenarios include mobile camera and moving targets in an uncontrolled environment [79], [93]. The results of [6] reveal that large motions along with full occlusions impede most of the trackers, thus even the linear motions with large target displacements in a scenario involving occlusion are considered as challenging for most of the tracker.

Inter-object interactions constitute another challenge which may cause the occlusion in videos. Partial occlusions, split-merge events (e.g., sports scenes with complex partial occlusions), grouping-fragmentation (e.g., a group of target pedestrians walking across a busy street), and full occlusions are typical scenarios emerged from target interactions.

Current occlusion datasets do not try to minimize the effects of other factors (scale variation, background clutter, etc.) in their occlusion scenarios and the attempts to simplify such videos (e.g., [78]) are very premature.

5. Conclusions and Future Directions

This study provided a comprehensive overview of the occlusion problem in online visual tracking. Based on the new categorization, occlusions are defined based on their three intrinsic attributes: duration, spatial extent, and complexity. Many studies tackled partial temporal occlusions, and significant progress has been made so that temporal partial occlusions with the spatial extent of 30% are considered as solved [6]. On the other hand, few attempts have been made to handle persistent or complex full occlusions, and even temporal full simple occlusions are still challenges for many trackers. This study collected the main problems caused by occlusion and provided the best practices in order to facilitate designing more robust trackers. By categorizing the state-of-the-art solutions to the occlusion problem, this study discussed the merits and demerits of each solution and illuminated the landscape for future research.

The thorough analysis in the survey highlighted effective approaches for robust tracking and provided potential future research directions in this field. Better foreground segmentation schemes, considering the split and merge events, dealing with varying number of objects, and incorporating other visual clues (such as context and motion patterns) into formulation of the association problem are recommended approaches to advance foreground trackers. By providing more robust detectors using latest breakthroughs in object categorization and fine-level object detection, the tracking-by-detection approaches would increase the accuracy of trackers. Feature detector and tracker fusion are trending solutions in this area while simultaneous localization and detection proved to be a successful strategy initiated by [34]. Moreover, part- and patch-based solutions and robust appearance models (locally sparse, discriminative, and occlusion-invariant) are the essence of recent successful trackers. Advanced motion models such as parametric models, stochastic sampling and adaptive motion models by the guidance of, e.g., optical flows or context information, seem to be another successful strategy. Cheap access to depth information provides the opportunity to use this rich source of information to disambiguate occlusion situ-

ations, to keep track of targets, to design powerful features, and to partially compensate the 3D-2D projection data loss. Robust detection of occlusions improves the tracker performance significantly and there are lots of work to do in this track. Utilizing hybrid models and switching mechanisms in order to compensate the demerits of different trackers with one another sounds promising. Occlusion reasoning is yet to be formalized, and by doing so, many ideas can be applied to this field. Formulating it as a competitive phenomenon between objects, decompositional approach and using context are a few directions in which preliminary studies gained success.

Dedicated evaluation frameworks and benchmarks to study off-the-shelf tracker under various occlusion cases promote research in this field. Further investigations and surveys on occlusion are required and databases covering all aspects and circumstances of occlusions are yet to be made. Additionally more detailed criteria provide more insights into the dynamics of the tracker during and after occlusion and help designing better trackers.

Acknowledgments

The authors are grateful to the reviewers and editor for their careful and close reading of the manuscript and fruitful comments. We would also like to thank Maryam Sadat Mirzaei for proofreading the manuscript.

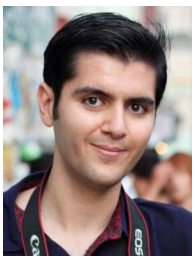
References

[1] P.F. Gabriel, J.G. Verly, J.H. Piater, and A. Genon, "The state of the art in multiple object tracking under occlusion in video sequences," Advanced Concepts for Intelligent Vision Systems, 2003.
[2] B.Y. Lee, L.H. Liew, W.S. Cheah, and Y.C. Wang, "Occlusion handling in videos object tracking: A survey," IOP Conference Series: Earth and Environmental Science, p.012020, IOP Publishing, 2014.
[3] P. Guha, A. Mukerjee, and V.K. Subramanian, "Formulation, detection and application of occlusion states (oc-7) in the context of multiple object tracking," AVSS'11, pp.191–196, IEEE, 2011.
[4] R. Vezzani, C. Grana, and R. Cucchiara, "Probabilistic people tracking with appearance models and occlusion classification: The ad-hoc system," PRL, vol.32, no.6, pp.867–877, 2011.
[5] K. Meshgi, S.I. Maeda, S. Oba, H. Skibbe, Y.Z. Li, and S. Ishii, "Occlusion aware particle filter tracker to handle complex and persistent occlusions," CVIU (submitted), 2014.
[6] A. Smeulders, D. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," PAMI, 2014.
[7] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," CVPR'13, pp.2411–2418, IEEE, 2013.
[8] O. Javed and M. Shah, "Tracking and object classification for automated surveillance," in ECCV'02, vol.2353, pp.343–357, Springer, 2002.
[9] B. Bose, X. Wang, and E. Grimson, "Multi-class object tracking algorithm that handles fragmentation and grouping," CVPR'07, pp.1–8, 2007.
[10] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "An adaptive color-based particle filter," IMVC, vol.21, no.1, pp.99–110, 2003.
[11] J. MacCormick and A. Blake, "A probabilistic exclusion principle for tracking multiple objects," IJCV, vol.1, pp.572–578, 1999.
[12] T.B. Dinh, N. Vo, and G. Medioni, "Context tracker: Exploring supporters and distracters in unconstrained environments," CVPR'11, pp.1177–1184, 2011.
[13] Y. Huang and I. Essa, "Tracking multiple objects through occlusions," CVPR'05, pp.1051–1058, IEEE, 2005.
[14] H.T. Nguyen and A.W. Smeulders, "Robust tracking using foreground-background texture discrimination," IJCV, vol.69, no.3, pp.277–293, 2006.
[15] F. Porikli, O. Tuzel, and P. Meer, "Covariance tracking using model update based on lie algebra," CVPR'06, pp.728–735, IEEE, 2006.
[16] D.A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," IJCV, vol.77, no.1-3, pp.125–141, 2008.
[17] J. Kwon and K.M. Lee, "Visual tracking decomposition," CVPR'10, pp.1269–1276, IEEE, 2010.
[18] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A.V.D. Hengel, "A survey of appearance models in visual object tracking," TIST, vol.4, no.4, 2013.
[19] I. Matthews, T. Ishikawa, and S. Baker, "The template update problem," PAMI, vol.26, no.6, pp.810–815, 2004.
[20] A.D. Jepson, D.J. Fleet, and T.F. El-Maraghi, "Robust online appearance models for visual tracking," PAMI, vol.25, no.10, pp.1296–1311, 2003.
[21] T. Zhang, B. Ghanem, C. Xu, and N. Ahuja, "Object tracking by occlusion detection via structured sparse learning," CVPRw'13, pp.1033–1040, IEEE, 2013.
[22] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting.," BMVC'06, pp.6.1–6.10, 2006.
[23] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in ECCV'08.
[24] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," CVPR'09, pp.983–990, 2009.
[25] S. Hare, A. Saffari, and P.H.S. Torr, "Struck: Structured output tracking with kernels," ICCV'11, pp.263–270, IEEE, 2011.
[26] S. Wang, H. Lu, F. Yang, and M.-H. Yang, "Superpixel tracking," ICCV'11, pp.1323–1330, IEEE, 2011.
[27] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan, "Locally orderless tracking," CVPR'12, pp.1940–1947, IEEE, 2012.
[28] J. Kwon and K.M. Lee, "Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive basin hopping monte carlo sampling," CVPR'09, pp.1208–1215, IEEE, 2009.
[29] X. Mei, H. Ling, Y. Wu, E.P. Blasch, and L. Bai, "Efficient minimum error bounded particle resampling l1 tracker with occlusion detection," IEEE Trans. Image Process., vol.22, no.7, pp.2661–2675, 2013.
[30] L. Čehovin, M. Kristan, and A. Leonardis, "An adaptive coupled-layer visual model for robust visual tracking," ICCV'11, pp.1363–1370, 2011.
[31] F. Yang, H. Lu, and Y.-W. Chen, "Bag of features tracking," ICPR'10, pp.153–156, IEEE, 2010.
[32] S. Zhang, H. Yao, X. Sun, and X. Lu, "Sparse coding based visual tracking: Review and experimental comparison," PR, vol.46, no.7, pp.1772–1788, 2013.
[33] F. Pernici, "Facehugger: The alien tracker applied to faces," ECCV'12 Workshops, vol.7585, pp.597–601, Springer, 2012.
[34] Z. Kalal, J. Matas, and K. Mikolajczyk, "Pn learning: Bootstrapping binary classifiers by structural constraints," CVPR'10, pp.49–56, 2010.
[35] M. Godec, P.M. Roth, and H. Bischof, "Hough-based tracking of non-rigid objects," CVIU, vol.117, no.10, pp.1245–1256, 2013.
[36] J. Fan, Y. Wu, and S. Dai, "Discriminative spatial attention for robust tracking," in ECCV'10, vol.6311, pp.480–493, Springer, 2010.
[37] J.H. Yoon, D.Y. Kim, and K.-J. Yoon, "Visual tracking via adaptive tracker selection with multiple features," in ECCV'12, vol.7575, pp.28–41, 2012.
[38] M. Spengler and B. Schiele, "Towards robust multi-cue integration for visual tracking," MVA, vol.14, no.1, pp.50–58, 2003.
[39] W. Choi and S. Savarese, "A unified framework for multi-target tracking and collective activity recognition," in ECCV'12, vol.7575, pp.215–230, 2012.
[40] J. Kwon and K.M. Lee, "Tracking by sampling trackers," ICCV'11,

pp.1195–1202, IEEE, 2011.
[41] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," IJCV, vol.56, no.3, pp.221–255, 2004.
[42] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," PAMI, vol.25, no.5, pp.564–577, 2003.
[43] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multi-target tracking," PAMI, vol.36, no.1, 2013.
[44] T. Bando, T. Shibata, K. Doya, and S. Ishii, "Switching particle filters for efficient visual tracking," RAS, vol.54, no.10, pp.873–884, 2006.
[45] J. Kwon, K.M. Lee, and F.C. Park, "Visual tracking via geometric particle filtering on the affine group with optimal importance functions," CVPR'09, pp.991–998, IEEE, 2009.
[46] M. Rodriguez, S. Ali, and T. Kanade, "Tracking in unstructured crowded scenes," ICCV'09, pp.1389–1396, IEEE, 2009.
[47] H. Grabner, J. Matas, L. Van Gool, and P. Cattin, "Tracking the invisible: Learning where the object might be," CVPR'10, pp.1285–1292, 2010.
[48] M. Yang, Y. Wu, and G. Hua, "Context-aware visual tracking," PAMI, vol.31, no.7, pp.1195–1209, 2009.
[49] H.T. Nguyen and A.W. Smeulders, "Fast occluded object tracking by a robust appearance filter," PAMI, vol.26, no.8, pp.1099–1104, 2004.
[50] N. Thome and S. Miguet, "A robust appearance model for tracking human motions," AVSS'05, pp.528–533, IEEE, 2005.
[51] A. Amer, "Voting-based simultaneous tracking of multiple video objects," Circuits and Systems for Video Technology, vol.15, no.11, pp.1448–1462, 2005.
[52] T. Zhao and R. Nevatia, "Tracking multiple humans in complex situations," PAMI, vol.26, no.9, pp.1208–1221, 2004.
[53] J. Sullivan and S. Carlsson, "Tracking and labelling of interacting multiple targets," in ECCV'06, pp.619–632, Springer, 2006.
[54] Z. Khan, T. Balch, and F. Dellaert, "Multitarget tracking with split and merged measurements," CVPR'05, pp.605–610, IEEE, 2005.
[55] A. Senior, A. Hampapur, Y.L. Tian, L. Brown, S. Pankanti, and R. Bolle, "Appearance models for occlusion handling," IMVC, vol.24, no.11, pp.1233–1243, 2006.
[56] A. Genovesio and J.C. Olivo-Marin, "Split and merge data association filter for dense multi-target tracking," ICPR'04, vol.4, pp.677–680, 2004.
[57] S.W. Joo and R. Chellappa, "A multiple-hypothesis approach for multiobject visual tracking," IEEE Trans. Image Process., vol.16, no.11, pp.2849–2854, 2007.
[58] S. Avidan, "Ensemble tracking," PAMI, vol.29, no.2, pp.261–271, 2007.
[59] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," PAMI, vol.32, no.9, pp.1627–1645, 2010.
[60] L. Zhu, Y. Chen, A. Yuille, and W. Freeman, "Latent hierarchical structural learning for object detection," CVPR'10, pp.1062–1069, 2010.
[61] S. Walk, N. Majer, K. Schindler, and B. Schiele, "New features and insights for pedestrian detection," CVPR'10, pp.1030–1037, 2010.
[62] M. Enzweiler, A. Eigenstetter, B. Schiele, and D.M. Gavrila, "Multi-cue pedestrian classification with partial occlusion handling," CVPR'10, pp.990–997, IEEE, 2010.
[63] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral channel features.," BMVC'09, pp.91.1–91.11, 2009.
[64] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," CVPR'06, pp.798–805, 2006.
[65] G. Duan, H. Ai, and S. Lao, "A structural filter approach to human detection," in ECCV'10, vol.6316, pp.238–251, Springer, 2010.
[66] W. Ouyang, X. Zeng, and X. Wang, "Modeling mutual visibility relationship in pedestrian detection," CVPR'13, pp.3222–3229, 2013.
[67] B. Wu and R. Nevatia, "Tracking of multiple, partially occluded humans based on static body part detection," CVPR'06, pp.951–958, 2006.
[68] D. Tang and Y.J. Zhang, "Combining mean-shift and particle filter for object tracking," ICIG'11, pp.771–776, IEEE, 2011.
[69] O. Javed, K. Shafique, Z. Rasheed, and M. Shah, "Modeling inter–camera space–time and appearance relationships for tracking across non-overlapping views," CVIU, vol.109, no.2, pp.146–162, 2008.
[70] Z. Duan, Z. Cai, and J. Yu, "Occlusion detection and recovery in video object tracking based on adaptive particle filters," Control and Decision Conference, 2009, pp.466–469, IEEE, 2009.
[71] S. Kwak, W. Nam, B. Han, and J.H. Han, "Learning occlusion with likelihoods for visual tracking," ICCV'011, pp.1551–1558, 2011.
[72] S. Song and J. Xiao, "Tracking revisited using rgbd camera: Unified benchmark and baselines," ICCV'13, pp.233–240, IEEE, 2013.
[73] Y. Wu, T. Yu, and G. Hua, "Tracking appearances with occlusions," CVPR'03, pp.I-789–I-795, IEEE, 2003.
[74] W. Hu, X. Zhou, M. Hu, and S. Maybank, "Occlusion reasoning for tracking multiple people," Circuits and Systems for Video Technology, IEEE Transactions on, vol.19, no.1, pp.114–121, 2009.
[75] X. Zhou, Y. Li, and B. He, "Game-theoretical occlusion handling for multi-target visual tracking," PR, vol.46, no.10, pp.2670–2684, 2013.
[76] A.S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," arXiv preprint arXiv:1403.6382, pp.512–519, 2014.
[77] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang, "Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol," PAMI, vol.31, no.2, pp.319–336, 2009.
[78] B.Y. Lee, L.H. Liew, W.S. Cheah, and Y.C. Wang, "Simulation videos for understanding occlusion effects on kernel based object tracking," in Computer Science and its Applications, vol.203, pp.139–147, 2012.
[79] A. Ess, B. Leibe, K. Schindler, and L. Van Gool, "A mobile vision system for robust multi-person tracking," CVPR'08, pp.1–8, 2008.
[80] S. Salti, A. Cavallaro, and L. Di Stefano, "Adaptive appearance modeling for video tracking: survey and evaluation," IEEE Trans. Image Process., vol.21, no.10, pp.4334–4348, 2012.
[81] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," CVPR'12, pp.1838–1845, 2012.
[82] B. Liu, J. Huang, L. Yang, and C. Kulikowsk, "Robust tracking using local sparse appearance model and k-selection," CVPR'11, pp.1313–1320, 2011.
[83] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," CVPR'12, pp.1822–1829, 2012.
[84] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust l1 tracker using accelerated proximal gradient approach," CVPR'12, pp.1830–1837, 2012.
[85] D.M. Chu and A.W.M. Smeulders, "Color invariant surf in discriminative object tracking," in ECCVw'10, pp.62–75, Springer, 2012.
[86] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai, "Minimum error bounded efficient l1 tracker with occlusion detection," CVPR'11, pp.1257–1264, 2011.
[87] H. Grabner and H. Bischof, "On-line boosting and vision," CVPR'06, pp.260–267, 2006.
[88] S. Stalder, H. Grabner, and L.V. Gool, "Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition," ICCVw'09, pp.1409–1416, 2009.
[89] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in ECCV'12, vol.7574, pp.864–877, Springer, 2012.
[90] P. Guha, A. Mukerjee, and K. Venkatesh, "Ocs-14: you can get occluded in fourteen ways," IJCAI'11, 2011.
[91] D.M. Chu and A.W.M. Smeulders, "Thirteen hard cases in visual tracking," AVSS'10, pp.103–110, 2010.
[92] B. Zhan, D.N. Monekosso, P. Remagnino, S.A. Velastin, and L.-Q. Xu, "Crowd analysis: a survey," MVA, vol.19, no.5-6, pp.345–357,

2008.

[93] W. Choi, C. Pantofaru, and S. Savarese, "A general framework for tracking multiple people from a moving camera," PAMI, vol.35, no.7, pp.1577–1591, 2013.



Kourosh Meshgi was born in 1984. He received his B.Sc in hardware engineering and M.Sc. in artificial intelligence from Amirkabir University of Technology, Iran in 2008 and 2010 respectively. He is currently a Ph.D. student in Graduate School of Informatics, Kyoto University. His research interest has been machine learning, computer vision, and robotics.

Shin Ishii received B.E., M.E., and Ph.D. degrees from the University of Tokyo in 1986, 1988, and 1997, and joined Ricoh Co Ltd. and ATR Human Information Processing Research Laboratories in 1988 and 1997. Currently, he is a professor with the Graduate School of Informatics, Kyoto University. His current interests are reinforcement learning, computational neuroscience, systems biology, and bioinformatics.