CHAPTER-4

OBJECT TRACKING

Visual tracking is popular due to its numerous applications in practice and diverse impacts in theory. An indispensable component of it is the inference of the motion parameters of some targets from video, e.g. the trajectories, scale and orientation, and joint pose configuration of the targets. The few typical applications are human-computer interaction, e.g. hand and face tracking for gaming, eye gaze tracking for disability assistance, security surveillance, e.g. Airport surveillance, door access control, and home monitoring, medical image processing, e.g. racking cardiac borders in MRI images or in echocardiography, multimedia applications, e.g. face and people tracking for video conferencing , lip tracking in audio-visual analysis, activity and event analysis, e.g. gesture tracking, facial expression tracking , robotics e.g. autonomous vehicle and intelligent traffic control.

This chapter gives a theoretical review of the area of object tracking. Section 4.1 gives an introduction of this concept. Section 4.2 discusses its basics including the commonly used feature in it. Section 4.3 acquaints with the various methods available for tracking. Lastly section 4.3 briefs the concept of occlusion.

4.1 Introduction

During past decades, there has been a rapid development in the research of visual tracking due to the fact of growth of computing power and the sharp drop of storage cost, prominently as video cameras become widely available. Numerous novel algorithms as well as a lot of classical algorithms were developed and applied to visual tracking. Like the Kalman filter [61], probabilistic data association filtering (PDAF) [62], multiple hypothesis tracking (MHT) [63], Bayesian inference on graphical models, particle filtering or sequential Mote Carlo [64] (also known as CONDENSATION in vision literature [65]), subspace analysis [66], kernel-based density estimation [67], variational analysis [68], and various machine learning algorithms, support vector machine (SVM) [69], relevance vector machine (RVM) [70] and on-line boosting [71].

Some other areas that benefited from and interacted with Object Tracking are local feature descriptor, object detection and recognition, image segmentation, and background modeling. Although in recent years there has been a remarkable advancement in both theory and practice, but visual tracking still remains a challenging task .The foremost question before us that what are those challenges. Most of the visual tracking algorithms are confronted by two slightly contradictory challenges:

- The demands for computational efficiency and
- The capability to handle the unpredictable variations of the targets.

Computational efficiency is an inherent constraint for tracking, since real-time processing is vital for the successes of most online applications and even for off-line video analysis applications due to the vast video data. Especially, when the motion parameters are in high dimensional space, it is time consuming to explore the large solution space. Without this computational constraint, tracking is no longer a stand-alone problem from detection and recognition tasks. The other fundamental challenge is the dynamic nature of the targets due to enormous and unforeseeable variations in real-world scenarios. In unconstrained environments, there are too many factors that may affect the evidence of the presence of targets, e.g. background may be cluttered or even contain some camouflage objects as distractions. Illumination conditions may change evenly or unevenly so as to affect the target appearance, moreover, partial occlusion, out-of-plane rotation, target deformation, and quick motion all may present severe threats to long-term robust tracking.

All these variations are unpredictable, and therefore it is extremely hard for a tracker to consider all the potential variations and identify target specific or non-specific image invariants in advance. Adding further complexity, the visual tracking algorithms have to deal with these variations in an unsupervised and incremental manner. After initialization, the trackers will have no supervision to verify the tracking results and can hardly discern whether the appearance of the target is changing or partial occlusion is happening, so the estimation error could be accumulated. Besides, it is expected that the trackers should be insensitive to inaccurate target initialization and low image resolution or poor quality. In summary, the demand for computational efficiency and the dynamic nature of the tracking scenario are the two core challenges that tracking algorithms need to focus on.

4.2 Basics of Object Tracking

Object Tracking enjoys an important place, due to various partly due to abundance of high-powered computers, the availability of high quality and inexpensive video cameras, and the increasing demands for automated video. There are three key steps in video analysis:

- Detection of interesting moving objects,
- Tracking of such objects from frame to frame, and
- Analysis of object tracks to recognize their behavior.

So, basically tracking can be defined as the problem of estimating the trajectory of an object in the image plane as it moves around a scene. A tracker assigns consistent labels to the tracked objects in different frames of a video. Additionally, depending on the tracking domain, a tracker can also provide object-centric information, such as orientation, area, or shape of an object. Major difficulties that tracking encounters are the complex object shapes, loss of information caused by projection of the 3D world on a 2D image, noise in images, complex object motion, non-rigid or articulated nature of objects, partial and full object occlusions, scene illumination changes, and real-time processing requirements.

Tracking problem can be simplified by imposing additional constraints on the motion or appearance of objects. For example, almost all tracking algorithms assume that the object motion is smooth with no abrupt changes. One can further constrain the object motion to be of constant velocity or constant acceleration based on a priori information. Prior knowledge about the number and the size of objects, or the object appearance and shape, can also be used to simplify the problem.

Numerous approaches for object tracking have been proposed which primarily differs from each other in the way it performs object representation that best suits its tracking applications, the types of image features it incorporates, the type of motion or, appearance models it uses. Such selection depends on the context in which tracking is performed and the end use for which the tracking information is being sought. A large number of tracking methods have been proposed which attempt to answer these questions for a variety of scenarios. Object Tracking is used in numerous tasks like:

– Human-Computer interaction: it is for the purpose of gesture recognition and for eye gaze tracking for data input to computers, etc; Microsoft Kinect is a motion sensing input device by microsoft for the XBOX 360 video game console and windows pcs. It enables users to control and interact with the XBOX 360 without the need to touch a game controller, through a natural user interface using gestures and spoken commands.

– Video indexing: It is the process of automatically assigning content-based labels to video documents like text indexing or bookmarking.

– Motion-Based Recognition: It deals with the human identification based on gait, automatic object detection, etc;

– Automated Surveillance: Under this task a scene is monitored to detect suspicious activities or unlikely events;

– Traffic Monitoring: Traffic monitoring is of great importance these days in real-time gathering of traffic statistics to direct traffic flow.

4.2.1 Features of Tracking

Features of an image are some special information that distinguishes it from the rest of the image. There could be many features present in a single image. Selection of a particular feature depends on the corresponding application. We need to perform various image transforms to extract its features. Choosing the right features plays a critical part in tracking. In general, the most desirable property of a visual feature is its uniqueness, so that the objects can be easily distinguished in the feature space. Feature selection is closely related to the object representation. For instance, object edges are used as features for contour-based representations, while, for histogram-based appearance, color is used as a feature. Often, a large number of tracking algorithms use a combination of these and some other features.

4.2.1.1 Color

In practice, the apparent color of an object is affected mainly by two physical factors namely the spectral power distribution of the light source, and the surface reflectance properties of the object. In image processing, the red-green-blue (RGB) color space is habitually utilized to represent color, even if it is not a perceptually uniform color space as the differences between colors in this space do not correspond

to the color differences perceived by humans and its dimensions are highly correlated. Others like L*u*v* and L*a*b* are perceptually uniform color spaces, while hue-saturation-value (HSV) is an approximately uniform color space, but these color spaces are sensitive to noise. In short, there is no final pronouncement on which color space is more efficient thus, a variety of color spaces have been used in tracking.

4.2.1.2 Edges

Boundaries of the objects present in the scene are called edges. Object boundaries typically cause strong changes in image intensities. Edge detection is utilized to recognize these changes. An important property of edges is that they are less sensitive to illumination changes, compared to color features. One of the famous algorithms that track the boundary of the objects is the canny edge detector.

4.2.1.3 Optical Flow

Optical flow contains a dense field of displacement vectors using which the translation of each pixel in a region is defined.. It is computed using the brightness constraint with the assumption of brightness constancy of corresponding pixels in consecutive frames. In simpler sense optical flow is used to assess motion between two frames or a sequence of frames, without any other prior knowledge about the content of those frames and the motion itself gives the indication of something interesting is going on. In the optical flow algorithm each pixel of the frame is associated with some kind of velocity or, equivalently, some displacement that represents the distance a pixel has moved between the previous frame and the current frame. Such an arrangement is usually referred to as a dense optical flow, which associates a velocity with every pixel in an image [74].

4.2.1.4 Texture

Texture measures the intensity variation of a surface and gives indication of the smoothness and regularity. As compared to color, texture requires a processing step to generate the descriptors and its features are less sensitive to illumination changes just similar to the edge features. Image texture features are generated via gray level co-occurrence matrix, run-length matrix, and image histogram. These are computed over gray levels, so for the application on the color images of the database these are first converted to 256 gray levels. Corresponding each image of the database, a set of texture features is

extracted. They are derived from a modified form of the gray level co-occurrence matrix over several angles and distances can be derived from a modified form of the run-length matrix over several angles and from the image histogram

4.2.1.5 Feature Selection

Depending on the application domain most of the features are chosen manually by the user. However, the automatic selection of features has received significant attention in the pattern recognition community. The methods for automatic feature selection can be classified into:

– Filter methods: In these methods a general criterion is used to select features based on general criteria, for instance the features should be uncorrelated. Principal component analysis is one example of it that involves transformation of number of correlated variables into a comparatively smaller number of uncorrelated variables.

– Wrapper methods: These methods choose the features based on their usefulness in a specific problem domain. For example, the AdaBoost (adaptive boosting) which is used to find a strong classifier based on a combination of moderately inaccurate weak classifiers: given a large set of features, one classifier can be trained for each feature. Then, a weighted combination of classifier (representing features) that maximize the classification performance of the algorithm will be discovered (the higher the weight of the feature, the more discriminatory it is) and the first n highest-weighted features are used for tracking. Among all features, color is one of the most widely used for tracking.

4.3 Methods for Tracking

Object tracking generates the trajectory of an object over time by discovering its exact position in every frame of the video. Object tracker may also provide the complete region in the image that is occupied by the object at every time instant. The task of detecting the object and establishing correspondence between the object instances across frames can either be performed separately by obtaining possible object regions in every frame by means of an object detection algorithm and then the tracker corresponds objects across frames or jointly in which the object region and correspondence is jointly estimated by iteratively updating object location and region information obtained from previous frame

In both of the tracking approaches the objects are represented using the shape like Points, Primitive geometric shapes, Object silhouette etc. The appearance models can be used like articulated shape models, Probability density of object appearance, Templates, Multi-view appearance models etc. Depending on the type of motion or deformation the model is selected to represent object shape, for instance: if an object is represented by a point, then only a translational model can be used. In the case where a geometric shape representation (like an ellipse) is used for the object, parametric motion models, like affine or projective transformations, are appropriate. These representations can approximate the motion of rigid objects in the scene. However for a non-rigid object, silhouette or contour is the most descriptive representation and both parametric and non-parametric models can be used to specify their motion. The major methods of object tracking are:

4.3.1 Point tracking

In point tracking the objects to be detected in consecutive frames are represented by points and the association of the points is based on the previous object state that may include object position and motion. An an external mechanism is required in this approach to detect the objects in every frame. Fig 4.1(a) shows the tracking scenario formulated as the correspondence of detected objects represented by points across frames. Point correspondence poses inefficiencies in the presence of occlusions, misdetections, entries and exits of objects. It can be divided into two broad categories:

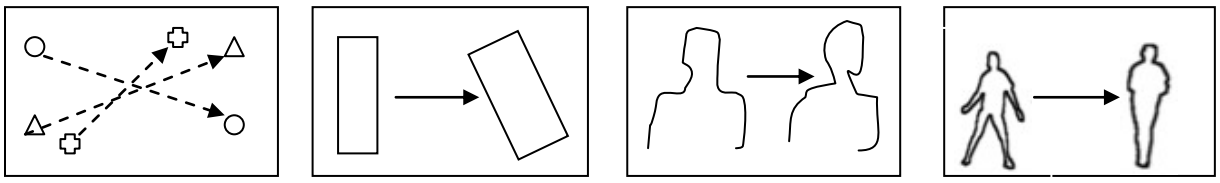


Fig 4.1 (a) Different tracking approaches. Multipoint correspondence, (b) parametric transformation of a rectangular patch, (c, d) two examples of contour evolution. [72]

i. Deterministic methods for correspondence

Under these methods a qualitative motion heuristics is used for constraining the correspondence problem. Deterministic methods for point correspondence define a cost of associating each object in

frame *t–1* to a single object in frame *t* using a set of motion constraints. It is aimed at minimizing the correspondence cost of the consecutive frames. Fig. 4.2(a) shows the all possible associations of points from one frame to another and Fig.4.2 (b) shows the one-to-one correspondences whereas Fig. 4.2(c) shows the Multi-view appearance models.

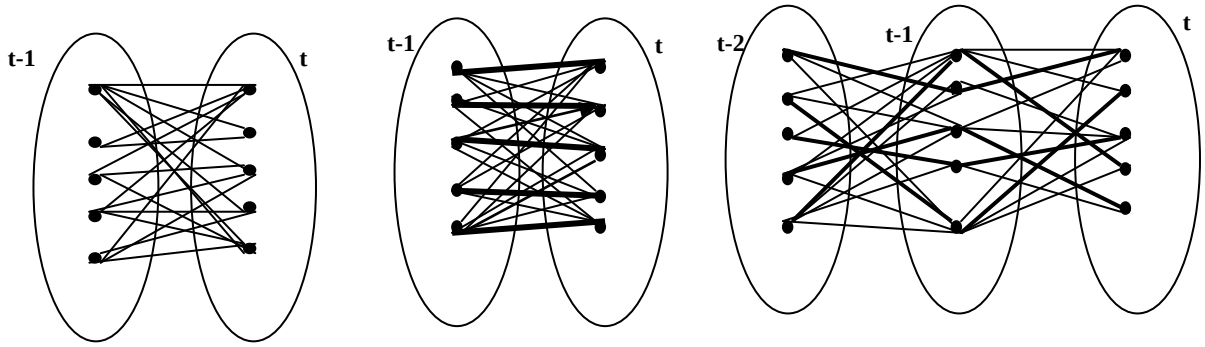


Fig. 4. 2 Point correspondence (a) All possible associations of a point (object) in frame *t* − 1 with points (objects) in frame *t*, (b) unique set of associations plotted with bold lines, (c) multiframe correspondences [72]

The correspondence cost is usually defined by using a combination of the following constraints:

Proximity assumes that the location of the object would not change notably from one frame to other [Figure 4.3 (a)].

Maximum velocity defines an upper bound on the object velocity and limits the possible correspondences to the circular neighborhood around the object [Figure 4.3 (b)].

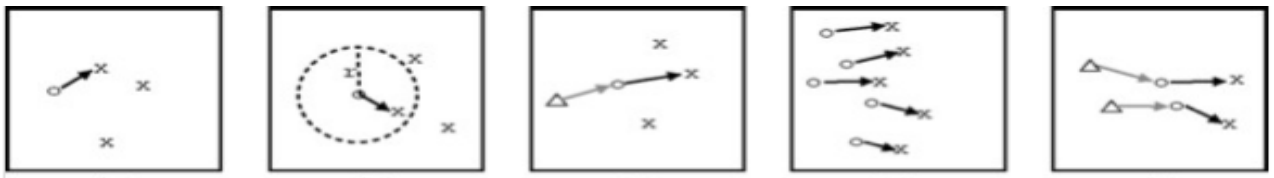
Small velocity change (smooth motion) assumes that the direction and speed of the object does not change drastically [Figure 4.3 (c)].

Common motion constraints the velocity of object in a small neighborhood to be similar [Figure 4.3(d)].

This constraint is suitable for objects represented by multiple points.

Rigidity assumes that objects in the 3D world are rigid consequently; the distance between any two points on the actual object will remain unchanged [Figure 4.3 (e)].

Proximal uniformity is a combination of the proximity and the small, velocity change constraints.



a) b) c) d) e)

Fig.4.3 Different motion constraints. (a) Proximity, (b) maximum velocity (*r* denotes radius), (c) small velocity-change, (d) common motion, (e) rigidity constraints. Δ Denotes object position at frame *t* − 2, ◦ denotes object position at frame *t* − 1, and finally × denotes object position at frame *t*.[72]

ii. Statistical methods for correspondence

Another name for Statistical methods is the probabilistic methods. These methods take into account the object measurement and uncertainties correspondence. In practice, measurements obtained from video sensors always contain noise. Besides it the object motion can be subjected to random perturbations like maneuvering vehicles. Statistical correspondence methods solve these tracking problems by considering the measurement and the model uncertainties during object state estimation. The statistical correspondence methods use the state space approach to model the object properties such as position, velocity and acceleration. Measurements usually consist of the position of the object in the image and are obtained by a detection mechanism. The information that represents a moving object in the scene (for instance its location) is defined by a sequence of states: X^t: t = 1, 2 . . . The change in state over time is governed by the dynamic equation:

$$\boldsymbol{X^t = f^t(X^{t-1}) + W^t} \quad (4.1)$$

Where $W^t = 1,2, \ldots$ is the white noise.

The relationship between the measurement and the state is specified by the measurement equation:

$$\boldsymbol{Z^t = h^t(X^t, N^t)} \quad (4.2)$$

Where N^t is the white noise and is independent of W^t. The objective of tracking is to estimate the state X^t , given all the measurements up to that moment. In other words, to construct the probability density function $\mathbf{p}\,(\boldsymbol{X^t | Z^{1,\ldots,t}})$. The optimal solution is provided by a recursive Bayesian filter, which solves the problem in two steps:

– The *prediction step,* which uses a dynamic equation and the already computed p.d.f (probability density function) of the state at time *t–1* to derive the prior p.d.f of the current state: $\boldsymbol{X^t | Z^{1,\ldots,t-1}}$).

– The *correction* step uses the likelihood function p($\boldsymbol{Z^t}|\boldsymbol{X^t}$) of the current measurement to compute the posterior p.d.f. $\boldsymbol{p}(\boldsymbol{X^t}|\ \boldsymbol{Z^{1,\ldots,t}})$. The measurements need to be associated to the corresponding object states.

There can be further two categories regarding the number of objects present in the scene that are to be tracked, i.e. a single object tracking or multiple object tracking is to be performed (MOT):

a) Single object state estimation

There can be two further possibilities in tracking a single object i.e either by employing kalman filters or, particle filters which are discussed as under:

– **Kalman filter:** If $\boldsymbol{f^t}$ and $\boldsymbol{h^t}$ are linear functions, and the initial state $\boldsymbol{X^t}$ and noise have a Gaussian distribution, then the optimal state estimate is given by the Kalman filter, which is composed of following steps:

Prediction. The prediction step uses the state model to predict the state of the variables:

$$\boldsymbol{X^t} = \boldsymbol{D}\boldsymbol{X^{t-1}} + \boldsymbol{W}, \tag{4.3}$$

$$\sum\nolimits^{\boldsymbol{t}} = \sum\nolimits^{\boldsymbol{t-1}}.\boldsymbol{D^T} + \boldsymbol{Q^t} \tag{4.4}$$

Where $\boldsymbol{X^t}$ and $\sum^{\boldsymbol{t}}$ are the state and covariance predictions at time *t*, $\boldsymbol{D}$ is the state transition matrix which defines the relation between the state variables at time *t* and *t*–1, $\boldsymbol{Q}$ is the covariance of the noise $\boldsymbol{W}$.

Correction. The correction step uses the current object observations $\boldsymbol{Z^t}$ to update the object's state.

– **Particle filters**: When the object state is not assumed to be a Gaussian, state estimation can be performed using particle filters. The extended Kalman filter (EKF) is a non-linear version of the Kalman filter and useful in the applications like navigation system and GPS. However, when the state transition f^t and observation models h^t are highly non-linear, the covariance is propagated through linearization of the underlying non-linear model and EKF can give poor performance. This problem can be dealt and solved by the unscented Kalman filter which uses a deterministic sampling technique, known as the unscented transform, to pick a minimal set of sample points, called sigma points, around the mean.

These sigma points are then propagated through the non-linear functions, from which the mean and covariance of the estimate are then recovered.

b) Multi-object data association and state estimation

Whenever the tracking of multiple objects is to be performed using particle or Kalman filters, there is a need of solving the correspondence problem. The correspondence problem brings about deterministic association between the most likely measurement for a particular object and its state. The simplest method to perform correspondence is using the nearest neighbor approach. However, if the objects are close to each other, then there is always a chance that the correspondence is incorrect. An incorrectly associated measurement can cause the filter to fail to converge.

4.3.2 Kernel tracking

Kernel tracking computes the motion of the object from one frame to the next. The motion of the object is in the form of parametric motion represented by a primitive object region like translation, conformal, affine, etc. in the form of dense flow field computed in subsequent frames. These algorithms differ in terms of the appearance representation used, the number of objects tracked, and the method utilized to estimate the object motion. These can be further categorized based on the appearance representation used.

4.3.2.1 Template and density-based appearance models

Templates and density-based appearance models are widely used because of their relative simplicity and low computational cost. It can be used for single object tracking as well for multiple object tracking.

(i) Single objects tracking

The template matching scheme is most common under it which uses a method for searching a similar region O_t in an image I_w. The O_t is defined from the previous frame.The position of the template in the current image is computed by using a similarity measure like cross correlation:

$$\boldsymbol{argmax}_{\boldsymbol{dx,dy}} \frac{\sum_x \sum_y [\boldsymbol{O_t}(\boldsymbol{x},\boldsymbol{y}) \times \boldsymbol{I_w}(\boldsymbol{x}+\boldsymbol{dx},\ \boldsymbol{y}+\boldsymbol{dy})]}{\sqrt{\sum_x \sum_y \boldsymbol{O_t^2}(\boldsymbol{x},\boldsymbol{y})}} \quad (4.5)$$

Where, (*dx, dy*) specifies the candidate template position.

Usually, image intensity or color features are used to form the templates. Since image intensity is very sensitive to illumination changes, image gradients can also be used as features. A major limitation of template matching is its high computation cost, due to the brute force search. To reduce the computational cost, the object search is usually limited to the vicinity of its previous position instead of performing a brute force search for locating the object by using a mean-shift procedure.

(ii) Multiple objects tracking

For multiple object tracking, the interaction between multiple objects and between objects and background is taken into account during the course of tracking. For instance the interaction between objects when one objects partially or completely occludes the other.

4.3.2.2 Multi-view appearance models

In the previous tracking methods, the appearance models, that is, histograms, templates etc., are usually generated online. Thus the models represented by it, are constructed from the information gathered about the object from the most recent observations. The objects may appear different from different views, and if the object view changes dramatically during tracking, the appearance model may no longer be valid, and the object track might be lost. To overcome this problem, different views of the object can be learned offline and used for tracking.

(i) Eigenspace

The subspace-based approach is discussed in [75] and referred to as eigenspace , to compute the affine transformation from the current image of the object to the image reconstructed using eigenvectors. First, a subspace representation of the appearance of an object is built using principal component analysis (PCA), and then the transformation from the image to the eigenspace is computed.

(ii) Support vector machine

In 2001, S.Avidan100 [69] used a support vector machine (SVM) classifier for tracking. SVM is widely used general classification scheme based on supervised learning. Given a set of positive and negative training examples, finds the best separating hyper plane between the two classes. During testing, the SVM gives a score to the test data indicating the degree of membership of the test data to the positive class. For SVM based trackers (SVT), the positive examples consist of the images of the object to be tracked, and the negative examples consist of all things that are not to be tracked. Generally, negative examples consist of background regions that could be confused with the object.

4.3.3 Silhouette tracking

Objects having complex shapes such as hands, head, and shoulders cannot be well described by simple geometric shapes. An accurate shape description of these objects is provided by a silhouette-based object tracker which aims at finding the object region in each frame by means of an object model generated using the previous frames. This object model can be in the form of a color histogram, object edges or the object contour. Silhouette trackers may be divided into two categories:

4.3.3.1 Shape matching

Shape matching is performed by searching for the object silhouette in the current frame. Shape matching can be performed using a process which is similar to tracking based on template matching, where an object silhouette and its associated model is searched in the current frame. The search is done by calculating the similarity of the object with the model generated from the hypothesized object silhouette based on previous frame. In this approach, the silhouette is assumed to be only translating from the current frame to the next, so non-rigid object motion is not explicitly handled. The object model usually in the form of an edge map, is reinitialized to handle appearance changes in every frame after the object is located. This update is required to overcome tracking problems related to viewpoint and lighting condition changes as well as non-rigid object motion. Another approach that can be utilized to match shapes is to find corresponding silhouettes detected in two consecutive frames.

4.3.3.2 Contour tracking

Contour tracking uses the state space models or direct minimization of some energy functional to evolve an initial contour to its new position in the current frame. Contour tracking methods, in contrast to the shape matching ones, iteratively evolve an initial contour in the previous frame to its new position in the current frame. This contour evolution requires that some part of the object, in the current frame, overlap with the object region in the previous frame.

4.3.4 Graph-based tracking

Graphs allow us to represent the structure in a rich and compact manner. Image graphs can be used to represent structure and topology. Node attributes set up like size, average color, position, edges are defined to specify the spatial relationships given by adjacency, border between the nodes. This way each image of a sequence is segmented and represented as a region adjacency graph. The graph-based tracking methods can be employed for associating structures acquired at different time instances. Object tracking then changes to particular graph-matching problem, in which the nodes representing the same object are to be matched. The intrinsic complexity of graph matching is considerably reduced by coupling it with the segmentation. Such methods are called as Methods using graph matching. If graphs are only used to represent structure, but not for associating consecutive measurements the methods come under the category of Methods not using graph matching. And, rest if the methods use graphs to represent task-specific prior knowledge in form of a graph structure.

4.4 Handling Occlusion

Occlusion is a very common phenomenon in any real time video. Categorically occlusion can be classified into four categories:

4.4.1 Self-occlusion

Self-occlusion occurs when one part of the object occludes another. This situation most frequently arises while tracking articulated objects.

4.4.2 Inter-object occlusion

Inter-object occlusion occurs when two objects being tracked occlude each other. Generally, for inter-object occlusion, the multi-object trackers can exploit the knowledge of the position and the appearance of the occluder and occludee to detect and resolve occlusion.

4.4.3 Background scene structure occlusion

Similarly, occlusion by the background occurs when a structure in the background occludes the tracked objects.

4.4.4 Combination of different kinds of occlusions

The change of object appearance problem arises when objects (for instance: human body) move in circle around their axis, changing their appearance.