

A Comparative Study of different Object Tracking Methods in a Video

Norah Almohaimed
Department of Computer Science
College of Computer
Qassim University

Master Prince
Department of Computer Science
College of Computer
Qassim University

ABSTRACT

Visual Object Tracking (VOT) is the most salient and an ongoing exploration field amongst the several disciplines of computer vision. The importance of this technology is due to the extensive range of applications such as robot navigation, human computer interaction, video surveillance, etc. The process of object tracking involves segmenting areas of a video scene and tracking its position, motion and occlusion. However, problems can appear during tracking on account of multiple issues including camera motion, object-to-object and object-to-scene occlusions, non-rigid structures, object and scene changes in patterns and appearance and abrupt object movement. The aim of this paper is to examine, analyze and provide a shortlist of the most ubiquitous object tracking techniques. This accomplish by providing a comprehensive review of the tracking process which involve object detection methods, object representation and features selection and object tracking over multiple frames. Object tracking methods are compared whilst elaborating upon the advantages and limitations.

General Terms

Visual Object Tracking

Keywords

Object detection, Object representation, Object tracking , Video surveillance

1. INTRODUCTION

Visual Object Tracking (VOT) is a relatively modern phenomenon occurred within the computer vision discipline. It is based on principles, which are linked to image or video frame processing. The data contained within this image or frame is captured and analyzed based on specific parameters, which allow an output to be generated accordingly. Two key steps facilitate this capability that include detection and representation of the object. Detection involves determining object existence whilst representation involves its detection[18]. Object tracking is therefore concerned with verifying the existence of an object within a video frame[19]. Object tracking can be defined as the estimation of object trajectory in the image plane as it moves from one frame to another. Tracking works through permanent label assignment of an object present in many video frames. Object tracking can be approached

in different ways based on factors such as suitability of object representation, specific use of image features, and object modeling parameters such as shape, appearance and motion. These factors are contingent upon the object and its environment. As a consequence, many tracking methods have been developed which take into consideration the aforementioned factors whilst stressing specific features over others [31]. The accuracy of tracking is contingent upon two factors including object detection and modeling. The flow diagram in Figure 1, illustrates the main steps associated with object tracking.



Fig. 1. Object tracking steps.

The first step of tracking objects in a video sequence involves object detection. This helps us to determine which object needs to be tracked followed by modeling presence of the object within a single frame. The last stage involves specifically focusing on, and tracking the desired object [12].

2. OBJECT DETECTION

Object detection involves determining semantic object instances of a specific class. For example, cars, boats and humans within digital videos and images [19]. The most important step in object tracking involves object detection mechanisms, which operate on a frame by frame basis or as the object first appears in a video. The most common method for detecting objects is to use single frame information. However, more advanced methods make use of temporal data computed from several frames in order to minimize the number of false detections. This temporal data typically exhibits itself as frame differencing evolving areas in sequential frames. Once regions of an object have been defined, a tracker is required to undertake the object's correspondence from one frame to the next for creating tracks [31]. Object detection can be broken down into three subcategories as shown in Figure 2.

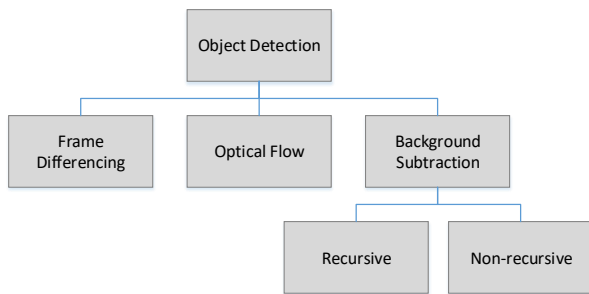


Fig. 2. Object detection methods [24].

2.1 Frame Differencing

This method is used in environments with static backgrounds with moving objects. In fact, objects of tracking interest are not stationary. They appear in different positions in consecutive frames [24]. It is possible to determine objects, which are moving by calculating the difference between two consecutive frames and isolating, where the movement is occurring. This form of detection is relatively easy and simple to carry out. Moreover, it can be applied to various environments which are dynamic in nature due to its adaptability. However, it can be difficult to acquire a complete outline of a moving object and could result in failed object detection or inaccurate results [25].

2.2 Optical Flow

The method which makes use of pattern of the apparent object motion and edges in video sequences based on relative motion between an observer and the scene is known as optic flow[15]. This method works by calculating the image optical flow field and carrying out clustering based on image optical flow distribution characteristics. This method is able to acquire complete movement data in addition to detecting the moving object. However, the main drawbacks to this method are it exhibits poor anti-noise capabilities, noise sensitivities, and requires a large number of calculations. These limitations restrict this method to non-real time applications [25].

2.3 Background Subtraction

It is possible to split a video sequence into complementary foreground and background. The object of interest is typically located within the image foreground, whilst the background is typically of negligible interest. Removal of background from a video frame leaves the foreground, that containing the object of interest. Background subtraction is more effective when the background is already known. For example, this method is highly applicable for surveillance cameras, which are motionless such as those found in road traffic surveillance. In this example, the road is perceived as being motionless which is advantageous, since the background has already been modeled. If the cameras are not motionless, then it needs to undertake background modeling prior to background subtraction with the aim of generating a reference model. This takes place by comparing a reference model and detecting the object based on variation in video sequences [24]. The difference between video frames and reference frame pixels indicate that an object is moving. Therefore, background modeling has to be capable of recognizing moving objects. Background subtraction can be implemented using simple algorithms, however, it may be sensitive to

changes in the exterior environment and is heavily impacted upon by interference [28]. It is possible to categorize background subtraction algorithms into recursive and non-recursive categories:

a) Non recursive methods
This method uses a sliding window approach to estimate the background by storing a buffer of the previous L video frames. It then estimates the background through temporal variation analysis of each pixel in the buffer [26]. Due to this fact, only limited frame numbers are held in the buffer, errors caused by frames outside the buffer limit are not considered. Non recursive techniques require very large amounts of storage due to the need for large buffers. However, this type of problem is typically mitigated by reducing video frame rates. Non recursive methods which are used commonly nowadays include non-parametric modeling, linear predictive filtering, median filter, and frame differencing [24].

b) Recursive methods
This method does not make use of a buffer for background estimation. Rather it updates a single background model recursively using an input frame. This means frames from a long time ago may impact the background model. Relatively speaking, recursive methods need less storage capacity, however, a background error may linger for a longer time [26]. Recursive methods which are commonly used nowadays include the mixture of Gaussians (MOG), Kalman filtering and approximated median filtering [24].

The main problem associated with background subtraction is spontaneously updating the background from each and every incoming video frame. This problem is compounded through problems such as bootstrapping, camouflage, shadows, memory, illumination changes and motion taking place in the background itself [28].

3. OBJECT MODELING

This plays a vital role in tracking because it categorizes an object of interest. Therefore, choosing the correct object model is critical for different applications. The object model defines the feature of interest and is used to estimate tracking. Object modeling has two features, the first is object representation, and the second includes the features used to characterize an object. Effective tracking is contingent upon the correct object model selection. The range of object representations takes into account different models which are best suited for slightly different circumstances. For example, certain circumstances require only simple models, whereas, other situations require complex object models to achieve effective tracking [12].

3.1 Object Representation

An entity which is intended to be tracked is termed as an object, and depending on the tracking application could be almost anything including people, vehicles, planes or birds etc. It is possible to represent an object by determining its size and shape [16]. The most common way of doing this is through the use of skeletal models, articulated shapes, contours, silhouettes, points and primitive geometric shapes such as ellipses and rectangles [31] as shown in Figure 3.

a) Point
Within object tracking, the point is the most trivial shape. The pixel location is used to represent an object by either using a specific characteristic of interest or object statistics such as the centroid. Point representation has been utilized within a broad

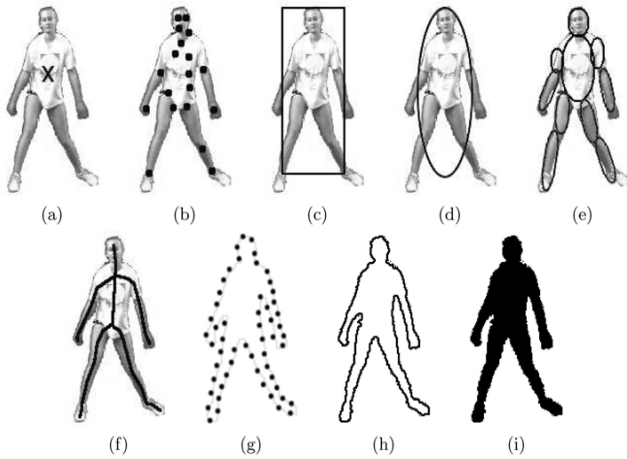


Fig. 3. Object representations[31]. (a) Centroid, (b) multiple points, (c) rectangular patch, (d) elliptical patch, (e) part-based multiple patches, (f) object skeleton, (g) complete object contour, (h) control points on object contour, (i) object silhouette.

spectrum of applications because of its processing simplicity, and the way points can be easily manipulated through algorithms [12]. It is possible to characterize an object relative to a set of points, which occupy a specific region of interest and is suitable for small area tracking applications [16].

b) Geometric shapes

An object can be represented based on a point in the simplest form. However, it does not totally cover the object dynamics. For example, rotation cannot be represented in this manner. This necessitates the use of more complicated parametric shapes. The most ubiquitous parametric shapes include those primitive geometric shapes such as circles, ellipses, squares, rectangles which are most suited for representing simple rigid objects. However, they can also be used for non-rigid objects after using adaptive methods. Geometric object tracking typically uses rectangle representation for applications such as car tracking, and in low distortion object tracking scenarios e.g. people[12].

c) Articulate shape models

This method can be utilized to track different object portions which are of interest. This permits segments of the object to be tracked individually such as a person who can be broken down into a head, torso and limbs. Hence, this model is most suited for objects which are made up of smaller components and can be represented using simple geometries such as ellipses, circles and rectangle. This model links constituent parts to a kinematic model [12].

d) Skeletal model

This can be applied to an object and to both rigid and articulated object types. A skeleton can be defined as a group of articulations that make up an object, which describe the dependencies and constraints within parts [12].

e) Object silhouette

This is also known as a blob and can be described as a non-disjoint, dense, binary mask which highlights an object. This method is most useful for pixel processing and it is possible to identify a moving object or a foreground by subtracting the background [12].

f) Contour

This method defines an object based on its contours and is advantageous over other methods due to its convenient non-parametric trade-off between storage needs and exhaustive object descriptors. Rather than complete silhouette storage, an object is described through the edges of its contours. This method can also represent non-rigid shapes efficiently [12].

Tracking is facilitated by representing a shape based on its appearance. From amongst the most ubiquitous appearance based methods, the following have been summarised:

a. Probability density appearance model

This method can model objects in both parametric and non-parametric means, i.e. Gaussian and a mixture of Gaussians, and histograms respectively. Object appearance features such as texture and color can be estimated using probability densities from the captured images [31].

b. Multiple-view object recognition

This method uses a set of views to model the object, where every view includes data concerning a small range of viewing condition. An object can be represented through two processes. The first involves clustering images, which represent different object views. The second is used when object information is widespread and used to form model view characteristics [16].

c. Active Appearance Models (AAMs)

These models consist of statistical data used to represent an object of interest via its grey level and shape appearance. These approaches take into account texture and shape which allow us to track the outline and appearance of an object simultaneously. These methods also permit the reusability of data provided through a tracker in a variety of other applications[12]. Despite this advantage, a training period is required by the system making use of AAM models. This can be undertaken using sample data, for example, the principal component analysis method[31].

d. Templates

These can be created using silhouettes or simple geometric shapes. One of the advantages of using this method is that templates carry both appearance and spatial data. However, this method is only suitable for objects which have limited variation[31].

3.2 Object Features

In the context of image processing, a feature can be understood as the simplest piece of data used for sorting out the computational task for a dedicated application. The feature typically is linked to a specific image structure and includes example such as texture, object, edges, etc. In demanding circumstances, a single feature may not provide the necessary detail. This necessitates the use of two or more features. The features can be used in order to differentiate the foreground object from the background object [16]. The use and selection of the precise features are vital when attempting to track an object. The visual uniqueness of an object is the most desirable property and facilitates easy recognition in the feature space.

The selection of the feature is strictly correlated to object representation, e.g. a histogram based appearance representation uses color as a feature. On the other hand, object edges are used as features for contour based representation. It is very common for tracking algorithms to use a combination of the aforementioned features[21]. Visual features which are considered to be the most common have been summarised below:

a) Color Modeling

Two physical factors influence the color of an object. Firstly this includes the spectral power distribution of the illuminate, and secondly the object surface reflectance attributes. During the acquisition of an image, the RGB color space is commonly used to display color. However, RGB cannot be considered as being perceptually uniform. In other words, there is a difference between the rendered RGB colors and those perceived by the human eye. This explains why color spaces such as LAB, YUV and HSV have been developed since they are more capable of representing color in a perceptually uniform manner. Despite this, colors can be negatively impacted because of noise. This explains why no single color space has been uniformly accepted since multiple color spaces have been used for different tracking purposes [23].

b) Shape Modeling

This can only be applied to the geometric shape of an object and cannot be applied to the structural analysis. It is contingent upon extracting geometry of regions such as boxes and ellipses which highlight object motion and thus facilitating classification. Through a variety of methods including using points, silhouettes, and boxes [16]. Edge is also used as a feature, where image intensities differ depending on object boundaries and can be detected using edge detection methods. The key feature of edges includes being minimally sensitive to illumination changes relative to color features. The majority of algorithms use Edge as the main agent in tracking objects [21].

c) Texture Modeling

The regular pattern and repetition of an element is termed texture and can be used to distinguish between different image areas and segment or classify different regions[5]. This object feature can be used for object classification as well as tracking and is used to locate the region of interest. Texture can be defined as a measurement of surface intensity variation, which is used to describe attributes such as regularity and smoothness. Unlike the color space model, determining texture requires the precursor step of descriptor generation [22]. Texture can be difficult to represent, but one of the ways in which texture can be modeled is through using two dimensional gray level variation. This collection of data permits pairs of pixels and their relative brightness to be computed based on contrast in direction, coarseness and regularity which provides the necessary data detect texture [5]. Relative to color, texture features are less sensitive to changes in light [21].

d) Motion Modeling

This method is based on the principle that object motion patterns and attributes are sufficient to distinguish it from other objects. For example, the motion of human beings has been shown to be distinct from other animals. Therefore, features such as motion can be used to recognize types of human movement, in addition to human identification [22]. This method is based on motion periodicity and can be made to learn how objects move and therefore categorize them effectively. This type of classification can be used with rigid and non-rigid objects. Although the tracking of rigid objects is easier because of motion periodicity. It is known that a limited amount of periodicity data occurs in non-rigid objects. This type of features also makes use of optical flow [24]. Regional pixel identification can be determined using a dense field of displacement vectors defined as an optical flow. Image features such as brightness can be taken as a measure during optical flow computations when corresponding pixels have the same brightness in consec-

utive frames. Optical flow feature is typically utilized in tracking applications and motion based object segmentation[5].

4. OBJECT TRACKING METHODS

Following object detection and correct modeling, the next step involves tracking. Object tracking can be defined as a process which tracks an object over a time period by locating its position in each video frame[9]. There is three main object tracking classifications including point, silhouette and kernel based tracking. In the case of point tracker detection, this takes place in every frame. On the other hand, kernel or geometric area based tracking and contour based tracking require object detection to take place only once as the object first appears [28]. Object tracking can be subdivided as shown in Figure 4.

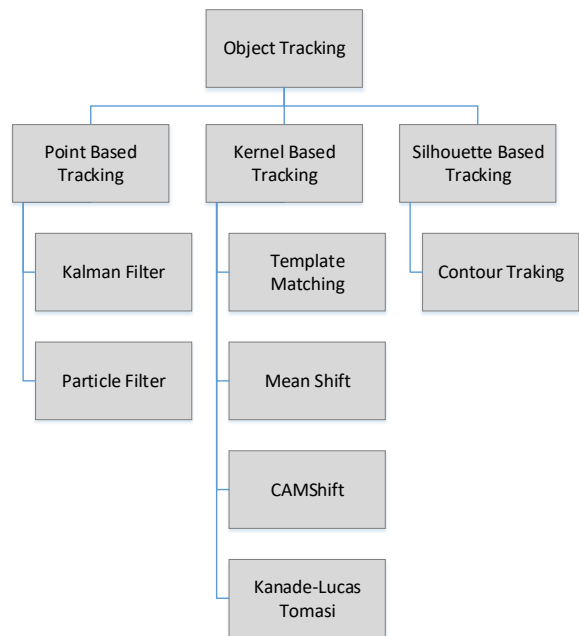


Fig. 4. Object tracking sub categories.

4.1 Point-based Tracking

This method represents an object using points and the correlation between these points are based on points of the previous frame. There are a wide number of algorithms available for point tracking purposes including Kalman Filter, Particle filter, etc. These methodologies are best suited for objects which can be simply taken as points. In the case of large objects, it is necessary to represent them using multiple points. One of the drawbacks of this method is that the presence of occlusions and misdetections cause points to become very complicated. Moreover, point tracking is unable to consider object entry and exit in the field of view [13].

a) Kalman Filter

This method is used to determine the linear system state where Gaussian distribution has taken place [31]. Tracking takes place by estimating the object position based on previous data

and object existence verification at the predicted location. Prior to carrying out tracking, the motion model must be learned by the system through the use of sample image sequences [17]. A simple way to describe this filter is by describing it as a set of mathematical equations which implement a predictor-corrector type estimator, where the optimal reading reduces the estimated error covariance under specific conditions [20].

Through the use of a feedback control form, the filter calculates the process state and then acquires feedback in the form of noisy measurements. As a consequence, the equations associated with this method fall into two subcategories, time update and measurement update equations. The time update equation is dedicated to projecting 'forward', the contemporary state and error covariance estimates to gain a priori estimates for the next step. The measurement update equation is dedicated to feedback. In other words, it incorporates a new measurement into a priori estimate to acquire an improved posterior estimate. The time update equation can also be considered as a predictor equation. On the other hand, the measurement update equation is to be considered as a corrector equation. The last estimation algorithm can be likened to a predictor-corrector algorithm which can be used for solving a number of problems [20], as demonstrated in Figure 5.

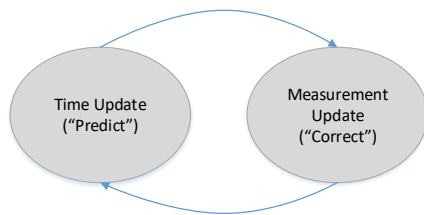


Fig. 5. Kalman filter cycle.

b) Particle filter

This method generates all the models for single variable prior to moving to the next variable. This method is advantageous when variables are dynamically generated and there are many variables. It also enables for another resampling to be undertaken. One disadvantage of using the Kalman filter is that state variables are assumed to be distributed using Gaussian methods. As a consequence, the Kalman filter makes poor estimations of non-Gaussian distribution variables. Particle filtering is one solution to this drawback [6].

This method is also known as the sequential Monte Carlo. It is from the most common approaches used to recursively construct the posterior PDF of the state space using Monte Carlo integration. The solution is typically applied to tracking problems by making use of the condensation algorithm. This type of filter makes use of algorithms which utilize color features, contours, or appearance models and can be described as being a Bayesian sequential sampling method. It works by recursively approximating the posterior distribution whilst using a limited number of weighted samples consisting of two steps including prediction and updating [3].

The underlying principle of particle filters involve being able to represent any pdf using a set of samples i.e. particles. One

set of values is associated with each particle for the state variables. Since this approach can be used to embody arbitrary distributions, then it is highly applicable to multi-modal and non-Gaussian pdfs. The function works by determining an approximate representation of a complex model, i.e. an arbitrary pdf instead of using a simplified model (i.e. Gaussians using exact representation). This method works based on the following:

1. The initial belief state is required.
2. This is followed by an iteration consisting of three phases, i.e. prediction, update and resampling.
 - a) Prediction. Every particle is taken and an additional sample from the motion model, which is added to it. This newly formed cloud of particles will hold the resultant position from the motion model. The prior distribution of particles can be used to approximate the resultant distribution.
 - b) Update. Sensor measurements are acquired, and each particle is assigned a specific weight, which is equal to a ratio of sensor measurement observation for the state of a particle. These weights are then subject to normalization. So the total sum to 1.
 - c) Resampling. Selection of new particles takes place to ensure every particle persists relative to its current weight. The highly unlikely particles are not selected at the fringe, whilst the highly likely particles are subject to replication at the cloud's center. This ensures that the high probability regions have a high density. Thus representing the posterior distribution more accurately [30].

4.2 Kernel-based Tracking

This method uses appearance and representations of the object of interest using rectangular or ellipsoidal shapes. Through the motion of each kernel on each frame, it is possible to track an object [11]. The motion of an object can be categorized in different ways including affine transformations, rotation and translation [24]. Several algorithms can be used for this purpose which differ depending on the quantity of object tracking, object representation and object motion estimation method. For example, in the case of real-time applications, it is common to represent objects using geometric shapes. One of the drawbacks of using geometric shapes is that they may not fully encapsulate the target object whilst background elements may also found therein [11]. Different methods can be used for kernel tracking including the mean shift method, CAMShift, simple template matching, and Kanade-Lucas-Tomasi (KLT) tracking. A summary of these methods has been provided below.

a) Mean shift tracking

This tracking method represents a target through density based appearance models. The algorithm used for this tracking method represents the appearance model using the histogram (texture, color etc) [2]. This method uses an iterative approach of tracking, which involves, determining similar pattern distributions located within a sequence of frames [4]. It is possible to improve target representation and accuracy by utilizing the chamfer distance transform method. The chamfer method is also capable of reducing the distance between two color distributions based on the Bhattacharya coefficient. This method can be characterized by discrete distribution of the sample and localized kernel.

The following steps are needed for kernel tracking:

1. Through the use of color features, it is possible to determine target probabilistic distribution in the first frame.
2. Distribution of the first and following frame can be compared.
3. Level of frame similarity can be determined using the Bhattacharya coefficient
4. Iterative loops will continue until the final frame [3].

In color based object tracking applications, the mean shift will be used due to its simplicity and robustness. Optimal results can be acquired whilst adhering to the following:

- One color is mainly used for the target object.
- There is no color changing of the target object.
- There are no automatic changes in target illumination.
- The scene does not contain objects which are similar to the target object.
- The background and the target object colors are different.
- The target object does not exhibit full occlusion [11].

The use of color histograms for target representation is a common phenomenon due to several advantageous characteristics including partial occlusion robustness, rotation and scaling independence [32]. Moreover, this method has relatively low computational overhead due to its simplicity. Despite these advantageous qualities, there are also several drawbacks to this method which include, an inability to change window scale when targets may be moving either away or towards the camera [8]. To mitigate this limitation, the continuous adaptive mean shift method has been forwarded (CAMShift) [10]. Moreover, Mean shift is incapable of tracking high speed objects within a frame [24]. This makes this method unsuitable for scenarios, where fast tracking of objects is required. However, this limitation can be partially mitigated using the mean shift algorithm together with the particle filter or Kalman filter [27].

b) Continuously adaptive mean-shift (CAMShift)

This method is built upon the principles found in mean shift and can be described as a lightweight and efficient tracking algorithm. The principal difference between mean shift and CAMShift is the latter uses continuously adaptive probability distributions. The main advantage of this approach is that it permits the probability distribution of a target to be recomputed in every frame. This permits the target appearance, shape and size to change in every frame. On the other hand, mean shift implements static probability distributions, which cannot be updated [1]. However, the main disadvantage of using this method occurs, when there are similarities between the background and the intended targets, which results in search window divergence [10]. This method is based upon the following iterations:

1. The image is separated into different components by setting the region of interest (ROI) for the probability distribution image.
2. Mean shift search window is selected, which defines the target distribution location intended for tracking.
3. The color probability distribution is calculated for the region that centered within the search window.
4. To determine the centroid of the probability image, the iterative mean shift algorithm is used. Both the central location and the zeroth moment (distribution area) can then be stored.
5. In the next frame, the mean location found previously in step 4 is used to re-enter the search window. This can be used to resize the window based on the zeroth moment function. Following this operation steps 3, 4 and 5 are iteratively repeated [1].

The algorithm used by CAMShift is flexible enough to track the distribution of any feature variant for the intended target in an efficient, robust and lightweight manner [7]. Although this flexibility exists, this is more common to find targets being represented using color data, since this provides practical performance and low complexity. This method is perhaps more suitable for objects which have a constant and simple appearance. Therefore, it is not suitable for scenes made up of high levels of complexity. Examples include scenarios, where the color distribution under background are similar to the target, or if the target shifts its location in front of different objects which consist of varying colors. In this situation, the probability of tracking an object effectively is greatly reduced [10].

c) Simple Template Matching

This method uses reference images for tracking purposes and is relatively simple in nature. It uses a template matching method for examining areas of interest and is therefore classed as a brute force method. This method uses a verified reference image with the frame which is separated from a video but partly overlaps an object. This method is used to process digital images in order to find a match in small parts of an image or an image that is equivalent in model in each frame. In the procedure, all potential positions of the template of the image of interest are considered and a numerical index is determined. That can be used to describe how effectively the model fits with the image position. This matching process takes place through the examination of every pixel in the image. In this case, the image of reference is acquired and then analyzed relative to successive video frames [3].

d) Kanade-Lucas-Tomasi (KLT) Tracking

Shi and Tomasi (1994) developed this method which is an extension of the work undertaken by Lucas and Kanade (1981). Due to its effectiveness and widespread use, it has gained an increasing amount of interest by researchers involved in motion tracking disciplines. The reasons why it is so appealing include the tracking algorithm itself, which allows features to be selected in an effective manner. KLT makes use of point features for tracking purposes which may include isolated pixels or corners, since they are easily distinguished and represent salient information for pinpointing the same features in following frames [29]. It is possible to acquire feature points for tracking purposes using several approaches. Of which the most common include this algorithm using the following: logic Eigen Features, SURF Features and Harris Corner Detector [17]. A simplified KLT algorithm is summarized as below:

1. In the first frame, the Harris corners are detected.
 2. The motion for each Harris corner is then calculated between successive frames.
 3. Each Harris point is tracked by linking motion vectors in each consecutive frame.
 4. Using Harris detection to create new Harris points based on m frames, i.e. between 10-15 frames.
 5. Utilize steps 1 to 3 to track old and new Harris points [14].
- The main advantages associated with this method include time efficiency, high accuracy and occlusion robustness. The method also can effectively manage the object's entry and exit points. By estimating the optical flow, it is possible to calculate the motion parameters of the moving objects and prevent the occlusion phenomenon of objects. The net sum of these features results in high tracking accuracy. However, tracking can falter even under normal conditions, when tracking multiple objects using this method [17].

4.3 Silhouette-based Tracking

Due to the complexity of many types of objects, they cannot be represented effectively using simple geometric shapes due to their irregularity, e.g. fingers and hands. Silhouette based tracking is one mean of mitigating this limitation. This method tracks an object in every frame based on the generation of an object model from previous frames [2]. This type of tracking method is typically used for an entire object has to be tracked.

The most advantageous feature of this method is that it can be used to manage a variety of objects having different shapes. It is possible to represent silhouettes in different ways. The most ubiquitous method makes use of a binary indicator function. This works by marking non-object regions with zeros and the object region with ones. Another aspect of silhouette tracking is occlusion handling. Silhouette tracker is also able to cope with split and merge of an object [31].

a) Contour tracking

This technique works by iteratively evolving an initial contour based on object location in previous frames to a new location in the current frame. The evolution of the contour is contingent upon object parts in the present frame overlapping the object in the prior frame. It is possible to use two different approaches, when tracking is based on contour evolution. The first method makes use of state space models, which are used for motion and contour shape modeling. The other method evolves the contour directly by reducing contour energy through direct minimization methods. For example, gradient descent. One of the most advantageous features of this method is its ability to manage object shapes of different varieties.

In this method, the silhouette can be represented implicitly through a function defined on a grid and can also be represented explicitly to describe its boundary via a collection of control points [31].

5. COMPARISON OF VIDEO TRACKING METHODS

A qualitative comparison for different tracking techniques that have been used for object tracking in a different situation is presented in Tables 1 and 2 that summarize the key features of different tracking methods including tracking object quantity, occlusion, and specific pros and cons. Whilst illumination does not impact point tracking, this method is in need of external mechanisms to detect objects in a series of frames. Problems associated with occlusion typically occur as objects entries or exits a frame and misdetections. However, Kernel and Silhouette tracking only require objects to be detected, as they first appear in a frame. Kernel tracking does not manage occlusion explicitly, and a major flaw of this method occurs because of object tracking acquired through primitive geometric representation. This can lead to sections of an object remaining outside the defined shape which could be filled with part of the background. This problematic issue can happen in the case of both rigid and non-rigid shapes. Silhouette tracking exhibits good flexibility when representing an object. Because it is capable of managing different shapes, including non-rigid shapes, even if they are complex in nature. Similar results have been reported in [31], [21] and [17] studies.

6. CONCLUSION

Object tracking and detection is a subset of computer vision discipline, which aims to determine the whereabouts of an object in

Table 1. Comparison for different object tracking methods. (S: single, M: multiple, P: partial.)

Tracking Methods	Type of tracking	Number of objects tracked	Occlusion Handling	Optimal
Kalman Filter	Point tracking	S	Yes	Yes
Particle Filter	Point tracking	M	yes	Yes
Mean shift	Kernel Tracking	S	P	No
CamShift	Kernel Tracking	S	No	No
KLT tracker	Kernel Tracking	S	Yes	Yes
Template Matching	Kernel Tracking	S	P	No
Contour Tracking	Silhouette Tracking	M	yes	Yes

Table 2. Advantages and limitations of object tracking methods.

Tracking Methods	Advantages	Limitations
Kalman Filter	Can track points in images which are noisy.	State variables are normally distributed (gaussian).
Particle Filter	Optimal results when evaluation of the image takes place at the hypothesis object position.	
Mean shift	Applicable for situations with dominant colors.	When the background is similar to the target, tracking problems arise.
CamShift	Resizable search window	Cannot be applied to complex scenes.
KLT tracker	Time efficient, robust occlusions.	Multiple object tracking becomes highly complex.
Template Matching	Relatively easier to implement and use.	Not suitable for complex templates. Problems can arise when objects temporarily leave the frame or become occluded.
Contour Tracking	Complex models for rigid and non-rigid objects can be handled. Illumination levels have minimal impact.	It is difficult to handle entry and exit of objects.

a series of video frames. In this paper the steps needed for object tracking have been documented. In view of the importance of object detection and modeling, several methods have been reviewed and discussed. A qualitative comparison has been done for different tracking techniques. Based on the acquired results, it is clear that occlusion is a persistent issue, and is arguably from amongst the most challenging matters with regards to object tracking. Therefore, researchers and developers need to continue improving algorithms for delivering more effective tracking results. This paper may be further developed in several ways. For example, algorithms may be combined to obtain better result in terms of speed, accuracy and occlusion. Also, it can be use a combination of features to acquire to broader benefits.

7. REFERENCES

- [1] John G Allen, Richard YD Xu, and Jesse S Jin. Object tracking using camshift algorithm and multiple quantized feature spaces. In *Proceedings of the Pan-Sydney area workshop on Visual information processing*, pages 3–7. Australian Computer Society, Inc., 2004.
- [2] J ATHANESIOUS and P Suresh. Implementation and comparison of kernel and silhouette based object tracking. *International Journal of Advanced Research in Computer Engineering & Technology*, pages 1298–1303, 2013.
- [3] J Joshan Athanesious and P Suresh. Systematic survey on object tracking methods in video. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 1(8):pp-242, 2012.
- [4] Bhakti Baheti, Ujjwal Baid, and Sanjay Talbar. An approach to automatic object tracking system by combination of sift and ransac with mean shift and klt. In *Advances in Signal Processing (CASP), Conference on*, pages 254–259. IEEE, 2016.
- [5] Vaijinath V Bhosle and Vrushsen P Pawar. Texture segmentation: different methods. *International Journal of Soft Computing and Engineering (IJSCE)*, 3(5):69–74, 2013.
- [6] Gary Bishop, Greg Welch, et al. An introduction to the kalman filter. *Proc of SIGGRAPH, Course*, 8(27599-3175):59, 2001.
- [7] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. ” O’Reilly Media, Inc.”, 2008.
- [8] Gary R Bradski. Computer vision face tracking for use in a perceptual user interface. 1998.
- [9] Joshua Candamo, Matthew Shreve, Dmitry B Goldgof, Deborah B Sapper, and Rangachar Kasturi. Understanding transit scenes: A survey on human behavior-recognition algorithms. *IEEE transactions on intelligent transportation systems*, 11(1):206–224, 2010.
- [10] Ebrahim Emami and Mahmood Fathy. Object tracking using improved camshift algorithm combined with motion segmentation. In *Machine Vision and Image Processing (MVIP), 2011 7th Iranian*, pages 1–4. IEEE, 2011.
- [11] Kaijen Hsiao, Jason Miller, and Henry de Plinval-Salgues. Particle filters and their applications. *Cognitive Robotics*, 4, 2005.
- [12] Anand Singh Jalal and Vrijendra Singh. The state-of-the-art in visual object tracking. *Informatica*, 36(3), 2012.
- [13] Kinjal A Joshi and Darshak G Thakore. A survey on moving object detection and tracking in video surveillance system. *International Journal of Soft Computing and Engineering*, 2(3):44–48, 2012.
- [14] Salil Kapur and Nisarg Thakkar. *Mastering OpenCV Android Application Programming*. Packt Publishing Ltd, 2015.
- [15] Upal Mahbub, Hafiz Intiaz, and Md Atiqur Rahman Ahad. An optical flow based approach for action recognition. In *Computer and Information Technology (ICCIT), 2011 14th International Conference on*, pages 646–651. IEEE, 2011.
- [16] Shipra Ojha and Sachin Sakhare. Image processing techniques for object tracking in video surveillance-a survey. In *Pervasive Computing (ICPC), 2015 International Conference on*, pages 1–6. IEEE, 2015.
- [17] divya s padmavathi S. Survey on tracking algorithms. *International Journal of Engineering Research & Technology (IJERT)*, 3, 2014.
- [18] Payal Panchal, Gaurav Prajapati, Savan Patel, Hinal Shah, and Jitendra Nasriwala. A review on object detection and tracking methods. *International Journal for Research in Emerging Science and Technology*, 2(1):7–12, 2015.
- [19] Himani S Parekh, Darshak G Thakore, and Udesang K Jaliya. A survey on object detection and tracking methods. *International Journal of Innovative Research in Computer and Communication Engineering*, 2(2):2970–2978, 2014.
- [20] Hitesh A Patel and Darshak G Thakore. Moving object tracking using kalman filter. *International Journal of Computer Science and Mobile Computing*, 2(4):326–332, 2013.
- [21] Sandeep Kumar Patel and Agya Mishra. Moving object tracking techniques: A critical review. *Indian Journal of Computer Science and Engineering*, 4(2):95–102, 2013.
- [22] Suraj Pramod Patil. Techniques and methods for detection and tracking of moving object in a video. 2015.
- [23] Fatih Porikli and Alper Yilmaz. Object detection and tracking. In *Video Analytics for Business Intelligence*, pages 3–41. Springer, 2012.
- [24] Kirubaraj Ragland and P Tharcis. A survey on object detection, classification and tracking methods. *Int. J. Eng. Res. Technol*, 3:622–628, 2014.
- [25] Rupali S Rakibe and Bharati D Patil. Background subtraction algorithm based human motion detection. *International Journal of scientific and research publications*, 3(5):2250–3153, 2013.
- [26] S Cheung Sen-Ching and Chandrika Kamath. Robust techniques for background subtraction in urban traffic video. In *Visual Communications and Image Processing 2004*, volume 5308, pages 881–893. International Society for Optics and Photonics, 2004.
- [27] A. Shivhare and V. Choudhary. Object tracking in video using mean shift algorithm: A review. *International Journal of Computer Science and Information Technologies*, 2015.
- [28] Gandham Sindhuja and MS Renuka Devi. A survey on detection and tracking of objects in video sequence. *International Journal of Engineering Research and General Science*, 3(2):418–426, 2015.
- [29] Supanee Tanathong and Impyeong Lee. The improvement of klt for real-time feature tracking from uav image sequence. *Dimension (pixels)*, 4288:2848, 2009.
- [30] Changjiang Yang, Ramani Duraiswami, and Larry Davis. Fast multiple object tracking via a hierarchical particle filter. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 212–219. IEEE, 2005.
- [31] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4):13, 2006.
- [32] Chunrong Zhang, Yuansong Qiao, Enda Fallon, and Chi-angqiao Xu. An improved camshift algorithm for target tracking in video surveillance. In *9th. IT & T Conference*, page 12, 2009.