Volume 3, Issue 11, November 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Comparison of Different Video Object Tracking Methods

Ritesh Singh* *PG student, Department of EXTC, D.J.Sanghvi C.O.E., Mumbai University, India*

Dr.Manali Godse *Professor, Department of BME, D.J.Sanghvi C.O.E., Mumbai University, India*

T.D.Biradar *Asst.Professor, Department of EXTC, D.J.Sanghvi C.O.E., Mumbai University, India*

Abstract: ***Object tracking finds its application in several computer vision applications, such as video compression, surveillance, robotics etc. Moving object detection and tracking are important steps in object recognition, context analysis and indexing processes for visual surveillance systems. It is a big challenge for researchers to make a decision on which tracking algorithm is more suitable for which situation or environment and to determine how accurately object is tracked (real-time or non-real-time).There is a variety of object detection and tracking algorithms (i.e. methods).In this survey, we categorize the tracking methods on the basis of the object and motion representations used, provide detailed descriptions of representative methods in each category, and examine their pros and cons.***

Keywords— Video, Particle, Kalman , Image, Tracking

I. INTRODUCTION

Tracking in videos is an important field of research since the new generations of computer processors can process more and more data. These researches are useful for the fields of surgery assisted by computer and robotics since they rely mostly on video processing to implement new devices. Usually, the video based object tracking deal with non-stationary images that change over time. Robust and Real time moving object tracking is a tricky issue in computer vision research area.Most of the existing algorithms is able to track only in predefined and well controlled environment. Some cases fail to consider the non-linearity issues. Particle filtering has proven to be very successful for non-Gaussian and non-linear estimation problems. In our system we are implementing particle filter to track a object in a video sequence and analyzing its advantages and disadvantages over other filters. Object tracking is an important task within the field of computer vision. The proliferation of high-powered computers, the availability of high quality and inexpensive video cameras, and the increasing need for automated video analysis has generated a great deal of interest in object tracking algorithms.

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.[1] Therefore, the use of object tracking is pertinent in the tasks of:
    1) Motion-based recognition, human identification based on gait, automatic object detection, etc;
    2) Automated surveillance, that is, monitoring a scene to detect suspicious activities or unlikely events;
    3) Video indexing, that is, automatic annotation and retrieval of the videos in multimedia databases;
    4) Human-computer interaction, that is, gesture recognition, eye gaze tracking for data input to computers, etc.
    5) Traffic monitoring, that is, real-time gathering of traffic statistics to direct traffic flow.
    6) Vehicle navigation that is, video-based path planning and obstacle avoidance capabilities.

II. BACKGROUND OF OBJECT TRACKING

In its simplest form, 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 based information, such as orientation, area, or shape of an object. Tracking objects can be complex due to: loss of information caused by projection of the 3D world on a 2D image, noise in images, complex object motion, nonrigid or articulated nature of objects, partial and full object occlusions, complex object shapes, scene illumination changes, and real-time processing requirements. [2]

Numerous approaches for object tracking have been proposed. These primarily differ from each other based on the way they approach the following questions: Which object representation is suitable for tracking? Which image features should be used? How should the motion, appearance, and shape of the object be modeled? The answers to these questions depend on the context/environment in which the 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. Our survey is focused on methodologies for tracking objects in general and not on trackers customized for specific objects.

III. A SURVEY OF OBJECT TRACKING

Almost all tracking algorithms require detection of the objects either in the first frame or in every frame. In Section 2, we will describe some common object shape representations, for example, points, primitive geometric shapes and object contours, and appearance representations. The next is the selection of image features used as an input for the tracker. In Section 4, we discuss various image features, such as color, motion, edges, etc., which are commonly used in object tracking. Almost all tracking algorithms require detection of the objects either in the first frame or in every frame. Section 5 summarizes the general strategies for detecting the objects in a scene. The suitability of a particular tracking algorithm depends on object appearances, object shapes, number of objects, object and camera motions, and illumination conditions. In Section 6, we categorize and describe the existing tracking methods and explain their strengths and weaknesses in a summary section at the end of each category.

IV. OBJECT REPRESENTATION

In a process of tracking, an object can be defined as anything that a person wants to track. It can be like, fish inside an aquarium, vehicles on a road, planes in the air, people walking on a road, or bubbles in the water are a set of objects etc. Objects can be represented by their shapes and appearances. [3] The object shape representations commonly employed for tracking are the following [11]:

- **Points:** The object is represented by a point, that is, the centroid (Fig. 1(a)), generally the point representation is suitable for tracking objects that occupy small regions in an image.
- **Primitive geometric shapes**: Object shape is represented by a rectangle, ellipse (Fig. 1(c), (d). Object motion for such representations is usually modeled by translation, affine, or projective transformation.
- **Object silhouette and contour**: Contour representation defines the boundary of an object (Fig 1(g), (h). The region inside the contour is called the silhouette of the object (see Fig 1(i)). Complex non-rigid shapes can be tracked by using Silhouette and contour representations
- **Articulated shape models**: Articulated objects are composed of body parts that are held together with joints as shown in Fig.1.

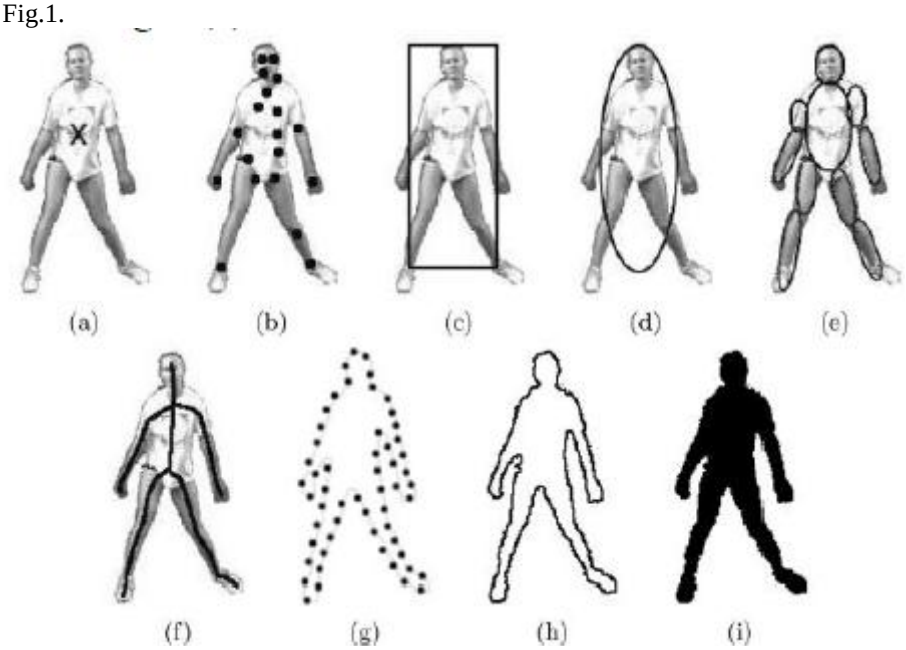

Fig.1 Object Representations from [3] a)centroid b)multiple points c)Regular patch d)elliptical Patch e)Part-Based Multiple Patches f)Object skeleton g)Complete Object Contour h)Control points on Object Contour i)Object Silhouette

- **Skeletal models**: Object skeleton can be extracted by applying medial axis transform to the object silhouette This model is commonly used as a shape representation for recognizing objects .Skeleton representation can be used to model both articulated and rigid objects.

In general, there is a strong relationship between the object representations and the tracking algorithms. Object representations are usually chosen according to the application. For the objects whose shapes can be approximated by rectangles or ellipses, primitive geometric shape representations are more appropriate. For tracking objects, which appear very small in an image, point representation is usually appropriate.

V. FEATURE SELECTION

Right feature selection plays a critical role 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 example, color is used as a feature for histogram-based appearance representations, while for contour-based representation, object edges are usually used as features. In general, many tracking algorithms use a combination of these features. Some common visual factors are:

- ***Color.*** : The apparent color of an object is influenced primarily by two physical factors, a) the spectral power distribution of the illuminant and b) the surface reflectance properties of the object. In image processing, the RGB (red, green, blue) color space is usually used to represent color. However, the RGB space is not a perceptually uniform color space, that is, the differences between the colors in the RGB space do not correspond to the color differences perceived by humans. In summary, there is no last word on which color space is more efficient, therefore a variety of color spaces have been used in tracking.
- ***Edges.*** Object boundaries usually generate strong changes in image intensities. Edge detection is used to identify these changes. An important property of edges is that they are less sensitive to illumination changes compared to color features. Algorithms that track the boundary of the objects usually use edges as the representative feature. Because of its simplicity and accuracy, the most popular edge detection approach is the Canny Edge detector.
- ***Texture.*** Texture is a measure of the intensity variation of a surface which quantifies properties such as smoothness and regularity. Compared to color, texture requires a processing step to generate the descriptors.

Mostly features are chosen manually by the user depending on the application domain. Automatic feature selection methods can be divided into *filter* methods and *wrapper* methods. The filter methods try to select the features based on a general criteria, for example, the features should be uncorrelated. The wrapper methods select the features based on the usefulness of the features in a specific problem domain, for example, the classification performance using a subset of features.[4]

VI. OBJECT DETECTION

Every tracking method requires an object detection mechanism either in every frame or when the object first appears in the video. A common approach for object detection is to use information in a single frame. However, some object detection methods make use of the temporal information computed from a sequence of frames to reduce the number of false detections. This temporal information is usually in the form of frame differencing, which highlights changing regions in consecutive frames. Given the object regions in the image, it is then the tracker's task to perform object correspondence from one frame to the next to generate the tracks.

We tabulate several common object detection methods in Table I. Although the object detection itself requires a survey of its own, here we outline the popular methods in the context of object tracking.[4]

Table I. Object Detection Categories

Categories	Representative Work
Point detectors	Moravec's detector [Moravec 1979], Harris detector [Harris and Stephens 1988], Scale Invariant Feature Transform [Lowe 2004]. Affine Invariant Point Detector [Mikolajczyk and Schmid 2002].
Segmentation	Mean-shift [Comaniciu and Meer 1999], Graph-cut [Shi and Malik 2000], Active contours [Caselles et al. 1995].
Background Modeling	Mixture of Gaussians[Stauffer and Grimson 2000], Eigenbackground[Oliver et al. 2000], Wall flower [Toyama et al. 1999], Dynamic texture background [Monnet et al. 2003].
Supervised Classifiers	Support Vector Machines [Papageorgiou et al. 1998], Neural Networks [Rowley et al. 1998], Adaptive Boosting [Viola et al. 2003].

VII. OBJECT TRACKING

The aim of an object tracker is to generate the trajectory of an object over time by locating its position in every frame of the video. The tasks of detecting the object and establishing correspondence between the object instances across frames can either be performed separately or jointly. In the first case, possible object regions in every frame are obtained by means of an object detection algorithm, and then the tracker corresponds to objects across frames. In the latter case, the object region and correspondence is jointly estimated by iteratively updating object location and region information obtained from previous frames. In either tracking approach, the objects are represented using the shape and/or appearance models described in Section 4.

For example, if an object is represented as 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. For a non-rigid object, silhouette or contour is the most descriptive representation and both parametric and nonparametric models can be used to specify their motion.

Below we represent the list of various tracking algorithms as defined in Table II.

Table II. Tracking Categories

Categories	Representative Work
Point Tracking	
• Deterministic methods	MGE tracker [Salari and Sethi 1990], GOA tracker [Veenman et al. 2001].
• Statistical methods	Kalman filter [Broida and Chellappa 1986], JPDAF [Bar-Shalom and Foreman 1988], PMHT [Streit and Luginbuhl 1994].
Kernel Tracking	
• Template and density based appearance models	Mean-shift [Comaniciu et al. 2003], KLT [Shi and Tomasi 1994], Layering [Tao et al. 2002].
• Multi-view appearance models	Eigentracking [Black and Jepson 1998], SVM tracker [Avidan 2001].
Silhouette Tracking	
• Contour evolution	State space models [Isard and Blake 1998], Variational methods [Bertalmio et al. 2000], Heuristic methods [Ronfard 1994].
• Matching shapes	Hausdorff [Huttenlocher et al. 1993], Hough transform [Sato and Aggarwal 2004], Histogram [Kang et al. 2004].

We now briefly introduce main tracking categories.

- ***Point Tracking:*** Objects detected in consecutive frames are represented by points, and the association of the points is based on the previous object state which can include object position and motion. This approach requires an external mechanism to detect the objects in every frame. An example of object correspondence is shown in figure 1(a).

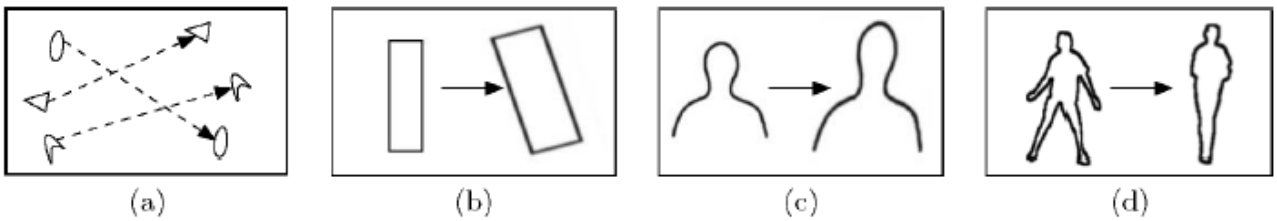

Figure 1. a) Different tracking approaches .Multipoint Correspondence, b)Parametric Transformation of a rectangular patch (c,d) Two examples of Contour Evolution

- ***Kernel Tracking***. Kernel refers to the object shape and appearance. For example, the kernel can be a rectangular template or an elliptical shape with an associated histogram. Objects are tracked by computing the motion of the kernel in consecutive frames (Figure 1(b)). This motion is usually in the form of a parametric transformation such as translation, rotation, and affine.
- ***Silhouette Tracking***. Tracking is performed by estimating the object region in each frame. Silhouette tracking methods use the information encoded inside the object region. This information can be in the form of appearance density and shape models which are usually in the form of edge maps. Given the object models, silhouettes are tracked by either shape matching or contour evolution (see Figure 1(c), (d)). Both of these methods can essentially be considered as object segmentation applied in the sequential domain using the priors generated from the previous frames.

7. A) STATISTICAL METHOD FOR POINT TRACKING

Measurements obtained from video sensors invariably contain noise. Moreover, the object motions can undergo random perturbations, for instance, maneuvering vehicles. Statistical correspondence methods solve these tracking problems by taking the measurement and the model uncertainties into account during object state estimation. The statistical correspondence methods use the state space approach to model the object properties such as position, velocity, and acceleration. [4] Measurements generally consist of the object position in the image, which is obtained by a detection mechanism.

Consider a moving object in the scene. The information representing the object, for example, location, is defined by a sequence of states $X^t : t = 1, 2, \ldots$. The change in state over time is governed by the dynamic equation,

$$X^t = f^t(X^{t-1}) + W^t$$

Where $W^t : t = 1, 2, \ldots$ is white noise. The relationship between the measurement and the state is specified by the measurement equation $Z^t = h^t(X^t, N^t)$, 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 or, equivalently, to construct the probability density function $p(X^t|Z^{1,\ldots,t})$. A theoretically optimal solution is provided by a recursive Bayesian filter which solves the problem in two steps. The *prediction* step uses a dynamic equation and the already computed pdf of the state at time $t-1$ to derive the prior pdf of the current state, that is, $p(X^t|Z^{1,\ldots,t})$. Then, the *correction* step employs the likelihood function $p(Z^t|X^t)$ of the current measurement to compute the posterior pdf $p(X^t|Z^{1,\ldots,t})$. In the case where the measurements only arise due to the presence of a single object in the scene, the state can be simply estimated by the two steps as defined. On the other hand, if there are multiple objects in the scene, measurements need to be associated with the corresponding object states. We now discuss the two cases.

- Single Object State Estimation
- Multi object Data Association and State Estimation.

7. A.1) Single Object State Estimation:

For the single object case, if f^t and h^t are linear functions and the initial state X^1 and noise have a Gaussian distribution, then the optimal state estimate is given by the Kalman Filter. In the general case, that is, object state is not assumed to be a Gaussian, state estimation can be performed using particle filters.[4]

❖ ***Kalman Filters:***

Kalman filter is used to estimate the state of a linear system where the state is assumed to be distributed by a Gaussian. Kalman filtering is composed of two steps, prediction and correction. The prediction step uses the state model to predict the new state of the variables:

$$\begin{aligned} \overline{X}^t &= \mathbf{D}X^{t-1} + W, \\ \overline{\Sigma}^t &= \mathbf{D}\Sigma^{t-1}\mathbf{D}^T + Q^t \end{aligned}$$

where X^t and Σ^t are the state and the covariance predictions at time t. $\mathbf{D}$ is the state transition matrix which defines the relation between the state variables at time t and $t-1$. Q is the covariance of the noise W. Similarly, the correction step uses the current observations Z^t to update the object's state:

$$\begin{aligned} K^t &= \overline{\Sigma}^t\mathbf{M}^T[\mathbf{M}\overline{\Sigma}^t\mathbf{M}^T + R^t]^{-1} \\ X^t &= \overline{X}^t + K^t\underbrace{[Z^t - \mathbf{M}\overline{X}^t]}_{v}, \\ \Sigma^t &= \overline{\Sigma}^t - K^t\mathbf{M}\overline{\Sigma}^t, \end{aligned}$$

where v is called the innovation, $\mathbf{M}$ is the measurement matrix, K is the Kalman gain. The Kalman filter has been extensively used in the vision community for tracking.

❖ ***Particle Filters.***

One limitation of the Kalman filter is the assumption that the state variables are normally distributed (Gaussian). Thus, the Kalman filter will give poor estimations of state variables that do not follow Gaussian distribution. This limitation can be overcome by using particle filtering. In particle filtering, the conditional state density $p(Xt|Zt)$ at time t is represented by a set of samples $\{s_t^{(n)} : n = 1, \ldots, N\}$ (particles) with weights $\pi_t^{(n)}$ (sampling probability). The weights define the importance of a sample, that is, its observation frequency.

The most common sampling scheme is *importance sampling* which can be stated as follows.

(1) Selection. Select N random samples $\hat{}s_t^{(n)}$ from $\mathbf{S}_{t-1}$ by generating a random number $r \in [0, 1]$, finding the smallest j such that $c_{t-1}^{(j)} > r$ and setting $\hat{} s_t^{(n)} = s_{t-1}^{(j)}$.

(2) Prediction. For each selected sample $\hat{} s_t^{(n)}$, generate a new sample by $s_t^{(n)} = f(\hat{}s_t^{(n)}, W_t^{(n)})$, where $W_t^{(n)}$ is a zero mean Gaussian error and f is a non-negative function, i.e. $f(s) = s$.

(3) Correction Weights $\pi_t^{(n)}$, corresponding to the new samples $s_t^{(n)}$ are computed using the measurements z_t by $\pi_t^{(n)} = p(z_t|x_t = s_t^{(n)})$, where $p(.)$ can be modeled as a Gaussian density.

Using the new samples $\mathbf{S}_t$, one can estimate the new object position by $\Sigma_n^N = 1\ \pi_t^{(n)} f\ s_t^{(n)}, W)$. Particle filter-based trackers can be initialized by either using the first measurements, $s_0^{(n)} \sim X_0$, with weight $\pi_0^{(n)} = 1/N$
or by training the system using sample sequences. In addition to keeping track of the best particles, an additional re-sampling is usually employed to eliminate samples with very low weights.

7. A.2) ***Multi object Data Association and State Estimation***.

When tracking multiple objects using Kalman or particle filters, one needs to deterministically associate the most likely measurement for a particular object to that object's state, that is, the correspondence problem needs to be solved before these filters can be applied. The simplest method to perform correspondence is to use 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. There exist several statistical data association techniques to tackle this problem. A detailed review of these techniques can be found in the book by Fortmann and Bar-

Shalom [1988] or in the survey by Cox [1993]. Joint Probability Data Association Filtering (JPDAF) and Multiple Hypothesis Tracking (MHT) are two widely used techniques for data association.

VIII. CONCLUSIONS

Point tracking methods can be evaluated on the basis of whether they generate correct point trajectories. Given a ground truth, the performance can be evaluated by computing *precision* and *recall* measures. In the context of point tracking, precision and recall measures can be defined as:

$$\text{precision} = \frac{\text{\# of correct correspondences}}{\text{\# of established correspondences}}$$

$$\text{recall} = \frac{\text{\# of correct correspondences}}{\text{\# of actual correspondences}},$$

Where, actual correspondences denote the correspondences available in the ground truth. Additionally, a qualitative comparison of object trackers can be made based on their ability to deal with entries of new objects and exits of existing objects, handle the missing observations (occlusion), and provide an optimal solution to the cost function minimization problem used for establishing correspondence.

Point trackers are suitable for tracking very small objects which can be represented by a single point (single point representation). Multiple points are needed to track larger objects. In the context of tracking objects using multiple points, automatic clustering of points that belong to the same object is an important problem. This is due to the need to distinguish between multiple objects and, between objects and background.

Following conclusions can be made.

- The Kalman filter is a recursive filter that estimates the state of a linear system at a given time from the estimated state from the previous time instant and the current measurement. It operates by computing a predicted value of the state and the covariance matrix of the estimation error, based on the estimate of the state vector at the previous time instant and the previously computed estimate error covariance matrix, and then updating them using the input available at that time instant..
- The extended Kalman filter or EKF is a Kalman filter that linearizes the original non-linear filter dynamics around the previous state estimates. It is a sub-optimal approach and is dependent on the noise being Gaussian distributed.
- Particle filters are recursive implementations of Monte Carlo based statistical signal processing. They approximate the optimal solution numerically based on a physical model, rather than applying an optimal filter to an approximate model. Hence Particle filters include a random element and almost surely converge to the true posterior pdf if the number of samples is very large. While the strong point of particle filters is that they can be used for non-Gaussian noise too.

ACKNOWLEDGMENT

Foremost, we would like to express our sincere gratitude to principal Dr. Hari Vasudevan for his continuous support to our study and research, for his patience, motivation, enthusiasm, and immense knowledge. Besides my advisor, I would like to thank our head of department Dr. Amit Deshmukh for his encouragement, insightful comments. Lastly, I would also like to thank my family for their constant support and time.

REFERENCES

[1] Kumar Jatoth, R., Shubhra, S., & Ali, E. (2013). Performance Comparison of Kalman Filter and Mean Shift Algorithm for Object Tracking. *International Journal of Information Engineering and Electronic Business*, *5*(5), 17–24. doi:10.5815/ijieeb.2013.05.03

[2] G, P. (2013). Video Object Tracking Using Particle Filtering, *2*(9), 2987–2993.

[3] Jacob, A. M., & Anitha, J. (2012). Inspection of Various Object Tracking Techniques, *2*(6), 118–124.

[4] Yilmaz, A., Javed, O., & Shah, M. (2006). Object tracking. *ACM Computing Surveys*, *38*(4), 13–es. doi:10.1145/1177352.1177355

