Survey on Algorithms for Object Tracking in Video

G. Lakshmeeswari
Asst. Professor, CSE
GITAM University

K. Karthik
Scholar, Dept. of CSE
Nagarjuna University

ABSTRACT
Object tracking is a very essential task in many applications of computer vision such as surveillance, vehicle navigation, autonomous robot navigation, etc. It contains detection of interesting moving objects and tracking of such objects from frame to frame. Its main task is to find and follow a moving object or multiple objects in image sequences. Normally there are three stages of video analysis: object detection, object tracking and object reorganization. This paper presents a brief survey of various video object tracking techniques like point tracking, kernel tracking and Silhouette tracking algorithms.

General Terms
Human-computer Interactions, 2D image, noise image, Proximity.

Keywords
Object tracking, point tracking, kernel tracking, silhouette tracking.

1. INTRODUCTION
Object tracking is an important task within the field of computer vision. 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. Object tracking is useful in the tasks of video indexing[1], Human-computer interaction, Traffic monitoring, etc.

Tracking can be defined as the problem of estimating the trajectory of an object in the image plane as it moves around a scene. Tracking objects may be complex due to reasons like: 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, Complex object shapes, Scene illumination changes and Real-time processing requirements.

1.1 Representation of Object
R E Kalman states that objects can be represented in three ways: Point tracking, Kernel tracking and Silhouette Tracking. Point tracking can be done using Deterministic and statistical methods[2]. Kernel tracking uses Template, density based appearance models and multi-view appearance models. Silhouette Tracking uses methods like contour evolution and matching shapes method[2].

2. LITERATURE SURVEY
2.1 Point Tracking
This can be formulated as the correspondence of detected objects represented by points across frames. Point correspondence is a complicated problem-specially in the presence of occlusions, misdetections, entries, and exits of objects. The deterministic methods use qualitative motion heuristics [3] to constrain the correspondence problem. On the other hand, probabilistic methods explicitly take the object measurement and take uncertainties into account to establish correspondence[2].

For point correspondence Deterministic Methods define a cost of associating each object in frame t−1 to a single object in frame t using a set of motion constraints. Minimization of the correspondence cost is formulated as a combinatorial optimization problem. The correspondence cost is usually defined by using a combination of constraints like Proximity, Maximum velocity, Small velocity change (smooth motion), Common motion, Rigidity and Proximal uniformity.

Correspondence is solved by a greedy approach based on the proximity and rigidity constraints by Sethi and Jain [4]. Their algorithm considers two consecutive frames and is initialized by the nearest neighbor criterion. The correspondences are exchanged iteratively to minimize the cost.

Finding trajectories: In a general but noise free case, elements may be moving in assorted directions. Suppose a feature detector gives points in each frame, the problem now is to find trajectories of m points in n frames. We can make general assumptions based only on motion characteristics not on nature of objects. The assumptions are :

1. An element in a frame can belong to one trajectory.
2. There should be m trajectories each containing n points
3. For each trajectory the deviation should be minimum
4. The sum of deviations of trajectories should be minimum

The algorithm proposed here is called greedy exchange algorithm. This algorithm will exchange trajectories up to (k+1)th frame, assuming that trajectories up to kth frame have already been calculated. The points of (k+1)th frame are assigned to the established trajectories using the nearest neighbors. The tentative trajectories are iteratively refined using this method. The value of criterion D for the tentative trajectories for the (k+1)th frame is as shown in Eq(1).

$$D = \sum_{p=1}^{m} D_p = \sum_{p==1}^{m} \sum_{q==2}^{s} d_p^q \quad \text{----------(1)}$$

Now we exchange the points from (k+i)th frame on i^{th} and j^{th} trajectories if we were to select only exchange from all possible exchanges, then we should make decision in favour of exchange maximizing the gain the exchange i and j if $g_{ij}^k > g_{rs}^k$ for all i,j,r,s. This idea leads to greedy exchange algorithm.

Each iteration step modifies at most 2 assignment pairs somewhere in the sequence, by exchanging the second entry of the pair. The algorithm considers all possible exchanges within the dmax range of two track heads in the whole sequence and the exchange that gives the highest gain by

decreasing the average criterion deviation is executed. The iteration phase stops when gain can no longer be obtained. The exchange gain between assignment pairs is defined in Eq(2):

$$g_{ij}^{k} Dc_{ip}^{k} Cc_{jq}^{k} .c_{iq}^{k} Cc_{jp}^{k} \text{ ----------(2)}$$

To achieve even better tracking results, the algorithm first optimizes correspondences over all frames in the forward direction and then (after this iteration phase stops) it optimizes correspondences in the backward direction. Only when the optimization process has not changed anything in either direction, the algorithm stops. This bi-directional optimization process can indeed increase the tracking quality, but, unfortunately, this process is not guaranteed to converge, especially with densely moving points.

A greedy approach is proposed by Rangarajan and Shah [6], which is constrained by proximal uniformity. Initial correspondences are obtained by computing optical flow in the first two frames. The method does not address entry and exit of objects. If the number of detected points decrease, occlusion or misdetection is assumed. Occlusion is handled by establishing the correspondence for the detected objects in the current frame. [2,6]. For the remaining objects, position is predicted basing on a constant velocity assumption. A non-iterative greedy algorithm is designed which assigns correspondence of points in one frame with the points in next frame. When minimum in the rows is considered it could happen that more than one minimum may lie along the same column j. That is, more than one point in frame k competes for point j in the (k+1)th frame. To get a one to one onto mapping, we should choose only one of these. However, this scheme should not just choose the minimum possible combination quantitatively, but it should prefer a combination where each individual correspondence is fairly good. The correspondence from frame k to k+1 involves m points. The minimum correspondence could be very favorable to some (m-1) points and not favorable for the m^{th} point. We should prefer a correspondence which is equally favorable to all points; at the same time we should not end up with a very high proximity path uniformity function[2,6].

Algorithm is

1. For *k* = 2 to *n* - 1 do

   (a) Construct M an ***(m*** * **n)** matrix, with the points from kth frame along the rows and points from ***(k***+1)th frame along the columns

2. Let m[i.j]=δ($X_p^{k-1}, X_i^k X_j^{k+1}$**),** where

$$\phi^{k-1}(p) = i \quad (4)$$

3. For a=1 to m do

   i) identify the minimum element in each row i of M

   ii) compute priority matrix B ,such that B[i, l_i] $= \sum_{i=1, j\neq l_i}^{m} M[i,j] + \sum_{k=1, k\neq i}^{m} M[k, l_i]$ ------(3)

   for each i,j

   iii) Select ***[i , l i]*** pair with highest **priority** value

   B[i,li], a nd make $\phi^k = l_i$

iv) Mask row ***i*** and column l_i from M

Veenman et al. [3], proposes a slightly modified version of Sethi and Jain [4] and Rangarajan and Shah [6] for matching object centroids, the objects are detected by using back-ground subtraction. The authors explicitly handle the change in the number of objects by examining specific regions in the image, for example, a door, to detect entries/exits before computing the correspondence. It proposes a mathematically rigorous methodology for tracking multiple objects. Two instantiations of the same tracking algorithm, with different initial conditions, are used to track two targets simultaneously. When one target passes close to the other, both tracking algorithms are attracted to the single target which best fits the head-and-shoulders model being used. We might think of avoiding this problem in a number of ways: interpreting the targets as "blobs" which merge and split again, enforcing a minimum separation between targets, or incorporating enough 3D geometrical information to distinguish the targets. However, each of these solutions can be unattractive[7].

2.1.1 Statistical Methods for CorrespondenceKalman Filters

The basic paper regarding the kalman filter is proposed by R. E. KALMAN in 1960 with the paper entitled "A New Approach to Linear Filtering and Prediction Problems". In his work he showed (i) Prediction of random signals; (ii) separation of random signals from random noise; (iii) detection of signals of known form (pulses, sinusoids) in the presence of random noise.

Kalman filter is a set of mathematical equations that provides an efficient computational (recursive) means to estimate the state of a process, in a way that minimizes the mean of the squared error. The filter is very powerful in several aspects it supports estimations of past, present, and even future states, and it can do so even when the precise nature of the modeled system. The Kalman filter is essentially a set of mathematical equations that implement a predictor-corrector type estimator that is optimal in the sense that it minimizes the estimated error covariance when some presumed conditions are met. In the prediction step, the time update is taken where one step ahead prediction is calculated (time update projects the current state estimate ahead in time. In the correction step, the measurement update is taken where the correction to the estimate of current state is calculated (the measurement update adjusts the projected estimate by a actual measurement at a time [8].

The equations for kalman filter fall into 2 groups: time update equations and measurement update equations[8,2]

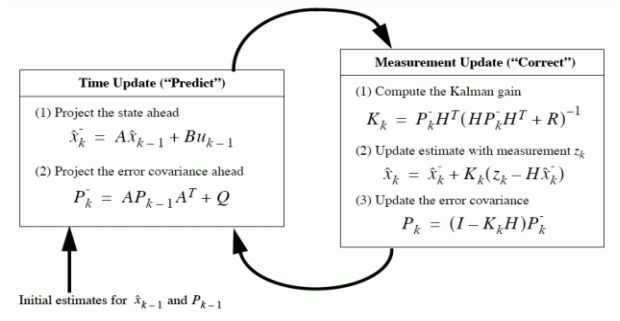

Fig 1 Time and measurement update equationsThe Kalman filter algorithm works as follows

Initialize **x**^0|0 and **P**0|0.

At each iteration *k*=1,…,*n*

Predict

$$X_{predicted} = AX_{n-1} + Bu_n$$

$$P_{predicted} = SP_{n-1}A^T + Q \quad \text{---------(5)}$$

Update

S=H$P_{predicted}\ H^T + R$

K=$P_{predicted}\ H^T S^{-1}$

$$X_n = X_{predicted} + Ky$$

$$P_n = (I - KH)P_{predicted} \quad \text{----------(6)}$$

Inputs:

U_n = Control vector. This indicates the magnitude of any control system's or user's control on the situation.

Z_n = Measurement vector. This contains the real-world measurement we received in this time step.

Outputs:

X_n = Newest estimate of the current "true" state.

P_n = Newest estimate of the average error for each part of the state.

Constants:

A = State transition matrix. Basically, multiply state by this and add control factors, and you get a prediction of the state for the next time step.

B = Control matrix. This is used to define linear equations for any control factors.

H = Observation matrix. Multiply a state vector by H to translate it to a measurement vector.

Q = Estimated process error covariance. Finding precise values for Q and R are beyond the scope of this guide.

R = Estimated measurement error covariance.

Kalman filter assumes a single measurement at each time instant, that is, the state of a single object is estimated. Tracking multiple objects requires a joint solution of data association and state estimation problems. Multi object Data Association and State Estimation. When tracking multiple objects using Kalman filter, it is needed 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 the converge.

2.2 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 This motion is usually in the form of a parametric transformation such as translation, rotation, and affine.[2]

Kernel tracking is typically performed by computing the motion of the object, which is represented by a primitive object region, from one frame to the next. The object motion is generally in the form of parametric motion or the 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 used to estimate the object motion. The tracking methods are divided into 2 subcategories based on the appearance representation used, namely, templates and density-based appearance models, and multi-view appearance models.

2.3.1 Tracking Using Template and Density-Based Appearance Models.

Templates and density-based appearance[9] models have been widely used because of Tracking single objects. The most common approach in this category is template matching. Template matching is a brute force method of searching the image, I_w , for a region similar to the object template, O_t defined in the previous frame. The position of the template in the current image is computed by a similarity measure, for example the their relative simplicity and low computational cost. We can divide the trackers in this category into two subcategories based on whether the objects are tracked individually or jointly cross correlation: arg

$$\max_{dx,dy} \frac{\sum_x \sum_y \left(o_t(x,y) + I_w(x + d_{x,y} + d_y)\right)}{\sqrt{\sum_x \sum_y O_T^2(X,Y)}} \quad \text{-----(7)}$$

where (d_x,d_y) specify 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 [3] can also be used as features.[2]

Instead of templates, other object representations can be used for tracking, for instance, colour histograms or mixture models can be computed by using the appearance of pixels inside the rectangular or ellipsoidal regions. Fieguth and Terzopoulos[1997[10] generate object models by finding the mean color of the pixels inside the rectangular object region. To reduce computational complexity, they search the object in eight neighboring locations. The similarity between the object model, M, and the hypothesized position,H, is computed by evaluating the ratio between the color means computed from M and H. The position which provides the highest ratio is selected as the current object location.

Mean Shift algorithm

Dorin Comaniciu nad Peter Meer[10] proposed the mean filter, which is an efficient approach to tracking objects whose appearance is defined by histograms

Mean shift is a procedure for locating the maxima of a density function given discrete data sampled from that function. It is useful for detecting the modes of this density. This is an iterative method, and we start with an initial estimate x. Let a kernel function $K(x_i - x)$ be given. This function determines the weight of nearby points for re-estimation of the mean. Typically a Gaussian kernel on the distance to the current estimate

$$K(x_I - x) = e^{-c\|X_i - X\|^2}$$

Is used. The weighted mean of the density in the window determined by K.

The mean-shift algorithm repeats the estimation until m(x) converges.

Strengths of the mean filter are

- Mean shift is an application-independent tool suitable for real data analysis,
- It Does not assume any predefined shape on data clusters
- It is capable of handling arbitrary feature spaces.
- It relies on choice of a single parameter: bandwidth**.**

Weaknesses of the mean filter are

- The selection of a window size is not trivial.
- Inappropriate window size can cause modes to be merged, or generate additional "shallow" modes.
- Often requires using adaptive window size.

Kanade–Lucas–Tomasi(KLT) feature tracker

The Kanade–Lucas–Tomasi (KLT) [13,14] feature tracker is an approach to feature extraction. It is proposed mainly for the purpose of dealing with the problem that traditional image registration techniques are generally costly. KLT makes use of spatial intensity information to direct the search for the position that yields the best match. It is faster than traditional techniques for examining far fewer potential matches between the images[2]

The KLT feature tracker is based on two papers: In the first paper, Lucas and Kanade[13] developed the idea of a local search using gradients weighted by an approximation to the second derivative of the image.

If h is the displacement between two images F(x) and G(x)=F(x+h) then the approximation is made as Eq (8)

F^l (x)~(F(x+h)-F(x))/F(x)=(G(x)-F(x)/h------(8)

$$h \approx \frac{G(x) - F(x)}{F^l(X)} \text{ ----------(9)}$$

This approximation to the gradient of the image is only accurate if the displacement of the local area between the two images to be registered is not too large. The approximation to $h$depends on x. For combining the various estimates of $h$at various values of x, it is natural to average them:

$$h \approx \frac{\sum_x \frac{G(x) - F(x)}{F^l(x)}}{\sum_x i} \text{ ----------(10)}$$

The average can be further improved by weighting the contribution of each term to it, which is inversely proportional to an estimate of $|F''(x)|$, where

$$F^{ll}(x) \approx \frac{G^l(x) - F^l(x)}{h} \text{ ----------(11)}$$

For the purpose of facilitating the expression, a weighting function is defined:

$$w(x) = \frac{1}{\left|G^l(x) - F^l(x)\right|} \text{ ----------(12)}$$

The average with weighting is thereby:

$$h = \frac{\sum_x \frac{w(x)[G(X) - f(X)}{F^l(x)}}{\sum_x w(x)} \text{ -------(13)}$$

Upon obtaining the estimate $F(x)$can be moved by the estimate of h. The procedure is applied repeatedly, yielding a type of Newton-Raphson iteration. The sequence of estimates will ideally converge to the best h. The iteration can be expressed by Eq(14)

$$h_0 = 0$$

$$h_{k+1} = h_k + \frac{\frac{w(x)[G(x) - F(x + h_k)]}{F^l(x + h_k)}}{\sum_x w(x)} \text{ ----(14)}$$

Generalization to multiple dimensions

The registration algorithm for 1-D and 2-D can be generalized to more dimensions[20]. To do so, we try to minimize the L_2 norm measure of error

$$E = \sum_{x \in R} [F(X = h) - G(x)]^2 \text{ -----(15)}$$

where $\mathbf{x}$and $\mathbf{h}$are n-dimensional row vectors. A linear approximation analogous to Eq(16):

$$F(x + h) \approx F(x) + h\left(\frac{\partial}{\partial_x} F(x)\right)^T \text{ -----(16)}$$

And partially differentiate $E$with respect to Eq(17):

$$0 = \frac{\partial_E}{\partial_h}$$

$$\approx \frac{\partial}{\partial_h} \sum_x \left[F(x) + h\left(\frac{\partial_F}{\partial_X}\right)^T - G(x)\right]^2$$

$$= \sum_x 2\left[F(x) + h\left(\frac{\partial_F}{\partial_x}\right)^T - G(x)\right]\left(\frac{\partial_F}{\partial_x}\right)$$

$$\Rightarrow h \approx \left[\sum_x [G(x) - F(x)]\left(\frac{\partial F}{\partial x}\right)\right]\left[\sum_x \left(\frac{\partial F}{\partial x}\right)^T \left(\frac{\partial F}{\partial x}\right)\right]^{-1}$$

----------(17)

In the second paper Tomasi and Kanade[14] improved the technique by tracking features that are suitable for the tracking

algorithm. The proposed features would be selected if both the eigenvalues of the gradient matrix were larger than some threshold.

By a very similar derivation, the problem is formulated as Eq(18)

$$\nabla d = e \text{ ----------(18)}$$

where ∇ is the gradient. This is the same as the last formula of Lucas–Kanade. A tracking method based on these two papers is generally considered a KLT tracker.

In the third paper, Shi and Tomasi[11] proposed an additional stage of verifying that features were tracked correctly. An affine transformation is fit between the image of the currently tracked feature and its image from a non-consecutive previous frame. If the affine compensated image is too dissimilar the feature is dropped. The reasoning is that between consecutive frames a translation is a sufficient model for tracking but due to more complex motion, perspective effects, etc. a more complex model is required when frames are further apart. Using a similar derivation as for the KLT, Shi and Tomasi showed that the search can be performed using the Eq(19)

$$Tz = a \text{-------(19)}$$

where T is a matrix of gradients, z is a vector of affine coefficients and a is an error vector. Compare this to $\nabla d = e$[2]

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. Both of these methods can essentially be considered as object segmentation applied in the temporal domain using the priors generated from the previous frames[9].

3 COMPARISON OF VIDEO TRACKING TECHNIQUES

Table 1 presents Comparison of video tracking techniques

Table 1: Comparison of Video Tracking Techniques

Type of tracking	**Methodology**	**Advantages**	**Limitations**
Point tracking	Sethi and Jain [1987][24] Finding trajectories of feature points in a monocular image sequence	Cost is low	Can't handle occlusions, entries, or exits etc.
Point tracking	MGE tracker	Can handle occlusions, entries, or exits etc..	Cost is high.
Point tracking	GOA Tracker	Handle occlusion and misdetection	Assume no object entries and exists
Point tracking	Rangarajan K, Shah M (1991) [13]Establishing motion corre-Spondence in the presence of occusion	Can handle occlusions	The method does not address entry and exit of objects.
Point Tracking	Kalman filter	Track points in noisy images	State is distributed by Gaussian
Kernel Tracking	Mean shift	1. Mean shift is an application-independent tool suitable for real data analysis. 2. Does not assume any predefined shape on data clusters. 3. It is capable of handling arbitrary feature spaces	1.The selection of a window size is not trivial. 2. Inappropriate window size can cause modes to be merged, or generate additional "shallow" modes. 3.Often requires using adaptive window size.
Kernel Method	Layering method	Suited for multiple object detection	Cost is high
Kernel Tracking	KLT Method	It is faster than traditional techniques for examining far fewer potential matches between the images.	Bare KLT less reliable than affine tracker and Feature trackers alone don't satisfy the needs of today AI applications.
Kernel Tracking	Eigen Tracking	Can hadle occusions,back ground clutter and noise etc.	Can not recognize more complex objects that change in both pition and view
Shilhuette tracking	State space models	Can handle complex models for rigid and non-rigid objects	Entry and exit of objects are difficult to handle, gestures recognition is not so accurate

4 CONCLUSION

Survey of tracking methods of categories like point tracking, kernel tracking and silhouette algorithms is presented. In the point tracking the methods namely Sethi and Jain method, Sethi method, Rangarajan and Shah method, Veenman et al., Kalman filter and Particle filter methods are studied. In the Kernel Tracking mean shift method, Kanade–Lucas–Tomasi(KLT) feature tracker method and Tao et al. layering methods are studied. In the Silhoutte method state space model is tracked. Comparative study of the above methods is carried out. The analysis shows that Kalman filter can track objects under noisy conditions also therefore this method can be optimal for object tracking in videos.

5 REFERENCES

[1] A. B lake, B. Basc le, M. Isard, J. MacCorm ick, "Statistical models of visual shape and motion", Phil. Trans. R. Soc. Lond. A (1998) 356, 1283–1302

[2] R. E. KALMAN "A New Approach to Linear Filtering and Prediction Problems", Transactions of the ASME–Journal of Basic Engineering, 82 (Series D): 35-45. Copyright © 1960 by ASME

[3] Veenman CJ, Reinders MJT, Backer E (2001) "Resolving motion correspondence for densely moving points". IEEE Trans on Pattern Analysis and Machine Intelligence 23(1):54–72

[4] Sethi IK, Jain R "Finding trajectories of feature points in a monocular image sequence". IEEE Trans on Pattern Analysis and Machine Intelligence 9(1):56–73

[5] Salari V, Sethi I "Feature point correspondence in the presence of occlusion". IEEE Trans on Pattern Analysis and Machine Intelligence 12(1):87–91

[6] K. Rangarajan, M. Shah. "Establishing motion correspondence", CVGIP: Image Understanding, 54:56–73, 1991

[7] Michael J. Black, Allan D. Jepson **"**Eigen tracking: Robust matching and tracking of articulated objects using a view-based representation". International Journal of Computer Vision 26(1), 63–84

[8] Prasad Kalane "Target Tracking Using Kalman Filter" International Journal of Science & Technology, ISSN (online): 2250,Vol. 2 Issue 2, April 2012

[9] Alper Yilmaz Omar, Mubarak Shah "Object tracking: A Survey", ACM Computing Surveys, Vol. 38, No. 4, Article 13, Publication date: December 2006.

[10] P Fieguth, D Terzopoulos. "Color - based tracking of heads and other mobile objects at video frame rates", Computer Vision and Pattern Recognition, 1997. Proceedings., IEEE

[11] 11. Jianbo Shi, Carlo Tomasi. "Good Features to Track", *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[12] Dorin Comaniciu "Mean Shift: A Robust Approach Toward Feature Space Analysis" IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 24, No.

[13] Bruce D. Lucas, Takeo Kanade. "An Iterative Image Registration Technique with an Application to Stereo Vision", *International Joint Conference on Artificial Intelligence*, pages 674–679, 1981

[14] Carlo Tomasi, Takeo Kanade. "Detection and Tracking of Point Features", *Carnegie Mellon University Technical Report CMU-CS-91-132*, April 1991.