Review on Tracking Object Motion in Video

Sandhya G. Alhat[1], Manasi K. Kulkarni[2]

[1]Department of Computer Engineering, PES's MCOE, Shivajinagar, Pune, India

[2]Professor, Departmentof Computer Engineering, PES's MCOE, Shivajinagar, Pune, India

Abstract: ***The process of locating a moving object and identifying its movements overtime is called as Visual Object Tracking. Tracking objects in motion in video sequences to improve recognition and tracking performances has evolved as a rapidly progressing research area in the domain of image processing and computer vision. This paper proposes a literature review of different target object representation methods and tracking techniques. The paper gives a survey of different tracking techniques like Local Steering Kernels (LSK), Particle Filter (PF), Kanade Lucas Tomasi (KLT) etc. It also describes the target object representation methods grouping them into five categories giving the significance of each.***

Keywords: visual object tracking;color histograms; local steering kernels; particle filter,kalman filter; kanade lucas tomasi

1. Introduction

Visual Object Tracking is the process of finding the location and dynamic configurations of one or more moving objects in each frame(image) of a video. Visual Tracking of an object in an image sequence is important for many applications, such as automatic video surveillance, autonomous robotic systems, human-computer interfaces, augmented reality, and e-healthcare. It can be a time consuming process as video contains huge amount of data. The task of detecting and tracking objects in motion is difficult to accomplish, as, in real life situations, the illumination conditions may vary and the object may be non-rigid , articulated , or occluded by background objects, and/or it may perform rapid and complicated movements, hence deteriorating tracking performance [1].In order to solve the above mentioned problems, numerous tracking algorithms have been proposed, which analyzes sequential video frames and outputs movement of target objects between frames. These algorithms employ techniques for object representation methods.

2. Target Object Representation Methods

Tracking Techniques employ different target object representation methods based on object features, texture and shape models, or object contours, object position prediction and searches the target object in the next video frame. There are five models of object representation : appearance based, model based, contour based, feature based and hybrid. Appearance based tracking methods use visual information of object projection like color, texture, shape etc. on image plane. These methods deal with simple object transformations like translation and rotation but sensitive to illumination changes [7]. Model based methods use prior information about object shape. These methods address problem of object tracking under illumination changes, change in object viewing angle and partial occlusion. But their computational cost is more. Also require implementation detailed model for each type in scene [8]. Contour-based tracking methods track object by considering their outline as boundary contours. These methods enable the tracking of both rigid and non-rigid objects[9]. Feature-based methods used to describe objects[11]. This process follows different steps as recognizing and tracking the object by extracting elements, to cluster elements in higher level features and to match these extracted features between images in successive frames. These methods perform well in partial occlusion and in tracking very small objects. The major drawback of feature based methods is the correct distinction between the target object and background features. Hybrid methods for object tracking use the advantages of the above mentioned methods, by combining two or more tracking methods[12] .Usually, feature-based methods are employed first, for object detection and localization. Then, region-based techniques are used to track its parts. The problem with these methods is their high computational complexity.

3. Literature Review

A. Local Steering Kernels(LSK) Object Tracking

The LSK Object Tracking technique makes the assumption that object translation and deformation between two successive video frames is small. Each transformation of the object image like scaling due to zooming or rotation is considered as an object instance. It is stored in a stack forming a list of object instances (images).

The stored object instances consist of the object model [6]. As tracking evolves, the object model is updated with new object instances, considering the transformations the object undergoes.

This paper tracks the target object in video frames based on the color histogram and the Local Steering Kernel (LSK) descriptor. It is based on appearance based object representation method. This framework takes as input the region of the target object in the previous video frame and a stored instance of the target object, and tries to localize the object in the current frame by searching the frame region that best resembles the input. As the object view changes over time, the object model is updated, hence incorporating these changes. Color histogram similarity between the detected object and the surrounding background is employed for background subtraction.
The LSK algorithm is as follows:

1. Initialization of the object region of interest (ROI) in the first video frame. The initialization can be done either manually, by selecting a bounding box around the object we want to track, or automatically, using an object detection algorithm, e.g. the one based on LSKs.
2. Color similarity search in the current search region, using CH information, which essentially leads to background subtraction and reduction of the number of the candidate object ROIs.
3. Representation of both the object and the selected search region through their salient features that are extracted using LSKs.
4. Decision on the object ROI in the new video frame, based on the measurement of the salient feature similarities between a candidate object ROI and: a) the object ROI in the previous frame, and b) the last stored object instance in the object model (stack) and finding a match [1].
5. Update the object model by storing its different views (called object instances) in a stack. When the match is successful, this update is done by pushing a new object instance in the stack, when the object undergoes an affine transformation, i.e., scale and rotation, or changes view.
6. Prediction of the object position in the following video frame and initialization of an object search region. The position prediction is based on the assumption that the object performs rather smooth motion [1].

B. Color Histogram

CH is sensitive to changes in illumination and view point. After predicting object position and selecting search region, the search region is divided into candidate object ROIs. The Color Histograms are been compared according to cosine similarity. It is an indicator of whether the search region patch belongs to the object ROI or the background. The cosine similarity among two histograms h1, h2€ R256 (Color Model) is given by

C (h1, h2) = Cos (Θ) = <h1, h2>/ ||h1|| ||h2|| (1)

Color histograms are the graphical representation of the colors which can be built for any kind of color space. The space is divided into an appropriate number of ranges, arranged as a regular grid, each containing many similar color values. A histogram of an image is been produced by discretization of the colors in the image into a number of bins, and counting the number of image pixels from each bin.

If images are multi-spectral, where each pixel is represented by an arbitrary number of measurements, the color histogram will be N-dimensional, where N being the number of measurements taken. Every measurement has its own wavelength range of the light spectrum, some of which may be outside the visible spectrum. If the set of possible color values is adequately small, every color may be placed on a range by itself; then the histogram is simply the count of pixels that have each possible color. Usually, the space is divided into ranges and often arranged as a regular grid, each containing many similar color values. The color histogram may also be represented and shown as a smooth function defined over the color space that minimizes the pixel counts.

C. Particle Filter Algorithm

Particle filtering is an inference technique for estimating the unknown motion state υ_t, from a noisy collection of observations, $Y_{1:t} = \{Y_1, ..., Y_t\}$ arriving in a sequential fashion. A discrete set of sample or particles represents the object- states and evolves over time driven. Non-linear motion models can be used to predict object-states. Particle Filter is concerned with the estimation of the distribution of a stochastic process at any time instant, given some partial information up to that point. A state space model is often employed to accommodate such a time series. Two important components of this approach are state transition and observation models whose most general forms can be defined as follows:

State transition model: $\upsilon_t = F_t (\upsilon_{t-1}, U_t$ (2)

Observation model: $Y_t = G_t(\upsilon_t, V_t)$ (3)

Where,
U_t is the system noise,
$F_t(., .)$ characterizes the kinematics,
V_t is the observation noise, and
$G_t(., .)$ models the observer.

The particle filter approximates the posterior distribution $p(\upsilon_t | Y_{1:t})$ by a set of weighted particles $\{\upsilon^{(j)}_t, w^{(j)}_t\}^J_{j=1}$. Zt is the image patch of interest in the video frame Y_t, parameterized by υ_t.

Particle Filter algorithm consists of 2 steps

1) Sequential importance sampling
It uses Sequential Monte Carlo simulation. For every particle at time t, we sample from the transition priors. For every particle we then evaluate and normalize the importance weights.

2) Selection step
It signifies to multiply or discard particles with respect to high or low importance weights w(j). This step allows us to track moving objects efficiently.

Particle Filter Theory:-

State Space:-
States are assigned as a location of target in each frame of the video.

System Dynamics
A second-order auto-regressive dynamics is chosen on the parameters used to represent our state space i.e. (x, y).
The dynamics is given as:Xt+1 = Axt + Bxt-1 Matrices A and B could be learned from a set of sequences where correct tracks have been obtained. We have used an ad-hoc model for our implementation.

Observation yt
The observation yt is proportional to the histogram distance between the color window of the predicted location in the frame and the reference color window.

Yt α Dist (q, q x),
Where q = reference color histogram.
qx = color histogram of predicted location.

Particle Filter Iteration [2]

Steps:

- Initialize xt for first frame.
- Generate a particle set of N particles $\{x^m\ t\}m=1..N$
- Prediction for each particle using second order autoregressive dynamics.
- Compute histogram distance.
- Weigh each particle based on histogram distance.
- Select the location of target as a particle with minimum histogram distance.
- Sampling the particles for next iteration.

```
Initialize a sample set $S0 = \{\upsilon_0^j, 1)\}^J_{j=1}$ according to prior distribution $p(\upsilon_0)$.
For $t = 1, 2, . . .$
  For $j = 1, 2, . . . , J$
  Resample $S_{t-1} = \{\upsilon^{(j)}_{t-1}, w^{(j)}_{t-1}\}$ to obtain a new sample $(\upsilon'^{(j)}_{t-1}, 1)$.
  Predict the sample by drawing $U^{(j)}_t$ for $U_t$ and computing $\upsilon^{(j)}_t = F_t(\upsilon^{(j)}_{t-1}, U^{(j)}_t)$.
  Compute the transformed image $Z^{(j)}_t$ .
  Update the weight using $w^{(j)}_t = p(Y_t|\upsilon^{(j)}_t) = p(Z^{(j)}_t|\upsilon^{(j)})$.
  End
Normalize the weight using $w^{(j)}_t = w^{(j)}_t / £^J_{j=1} w^{(j)}_t$
End[2]
```

D. Kalman Filter

The Kalman filter [13] is also known as linear quadratic estimation (LQE). It is an algorithm that uses a series of measurements observed over time. Measurements can contain noise (random variations) and other inaccuracies, and it also produces estimates of unknown variables that are more precise The Kalman filter operates recursively on streams of noisy input data and produce a statistically optimal estimate of the underlying system state.

$$K_t = \hat{P}_t H^T (H P_t^T H^T + R)^{-1} \tag{4}$$

$$\hat{x}_t = \hat{x}_{t-1} + K_t (z_t - H\hat{x}_t) \tag{5}$$

$$P_t = (I - K_t H) P_t \tag{6}$$

The matrix K_t is called the Kalman gain and is chosen such that minimizes the a posteriori error covariance P_t. The object will then be searched in a search region centered at the predicted position x_t. The size of this region varies according to the expected maximal object velocity, the object size and the reliability of the predicted position[15].

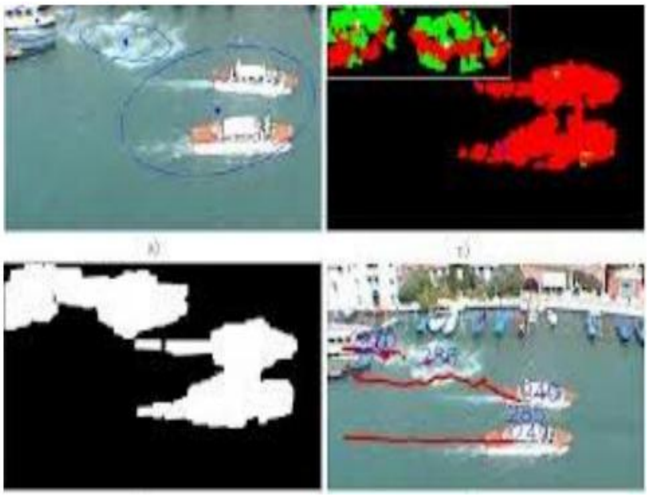


Figure 1: Kalman Filter Tracking

E. Kanade Lucas Tomasi(KLT)Tracking

The KLT algorithm is a feature detection and tracking process.

Feature tracking: Sum of Squared Difference (SSD) criteria applied to find the feature point whose window minimizes the following energy function:

$$E_t(dx, dy) = \sum [I(x+dx, y+dy, t+dt) - I(x, y, t)]^2 \tag{7}$$

Feature quality: If the quality of a tracked feature point decreases below a chosen threshold, that point is removed from consideration. To compensate, new features are identified in the same window. The feature point with the minimum criteria is retained if its SSD is below a certain threshold.

The KLT algorithm can be divided into two main parts. During the detection process, salient feature points are found and added to the already existing ones. Afterwards, in the tracking process for each feature point its corresponding motion vector is calculated. In the following, we describe each part of it in more detail.

Note that in the following Greek letters denote scalars, lowercase letters denote column vectors and uppercase letters denote matrices. We denote *I* as the current image and *J* as the immediately next image in the sequence. We write $\nabla I = \partial I / \partial(x, y)$ as the spatial image gradient of *I*, which is typically done with the Sobel or Sharr operator for robustness. Also we define *W(p)* as a small rectangular region centered at a given point *p*. Typically *W(p)* will be a 5 x 5 or 7 x 7 pixel neighbourhood. As the tracking is done with sub-pixel precision, p will have non-integer coordinates. Its neighbours are then calculated using bilinear interpolation.

Feature Point Detection

The task here is to detect new feature points in a given image *I* and add them to the already existing feature points. In order to track feature points reliably, their pixel neighbourhood should by richly structured. As a measure of 'structuredness' of the neighborhood of a pixel *p*, one can define the structure matrix *G*:

$$G = \sum_{x \in W(p)} \nabla I(x) \cdot \nabla I(x)^T$$

Its eigen values λ1, λ2 (which are guaranteed to be ≥ 0 as the matrix is positive semi-definite) deliver useful information about the neighborhood region *W*. If *W* is completely homogenous, then λ1 = λ2 = 0. In contrast, λ1 > 0, λ2 = 0 indicates that *W* contains an edge and λ1 > 0, λ2 > 0 indicates a corner. The smaller eigen value λ = min(λ1, λ2) can now be used as a measure of the cornerness of *W*, where larger values means stronger corners.

The feature detection is now composed of the following steps:

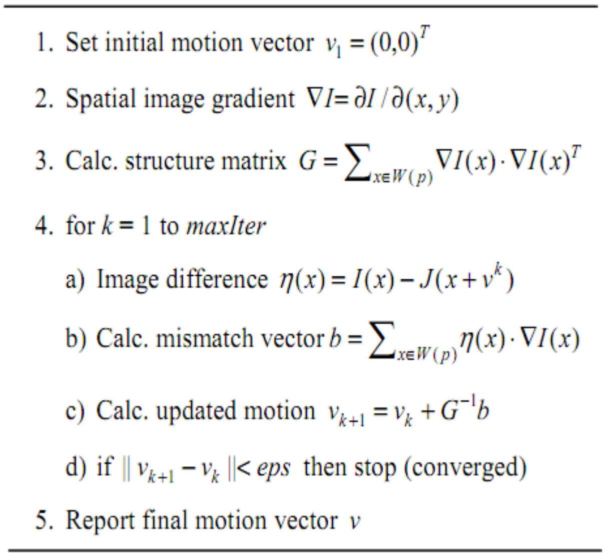
1) Calculate structure matrix *G* and cornerness λ for each pixel in the image *I*.
2) Calculate the maximum cornerness λ_{max} occurring in the image.
3) Keep all pixels that have a cornerness λ larger than a certain percentage (5% - 10%) of λ_{max}.
4) Do a non-maxima suppression within the 3 x 3 pixel neighbourhood of the remaining points to keep only
5) the local maxima.
6) From the remaining points, add as many new points to the already existing points as needed, starting with the points with the highest cornerness values. To avoid points concentrated in some area of the image, newly added points must have a specific minimum distance (e.g. 5 or 10 pixels) to the already existing points as well as to other newly added points (*Minimum-Distance-Enforcement*).

Feature Point Tracking

In the tracking step, we want to calculate for each feature point *p* in image *I* its corresponding motion vector *v* so that its tracked position in image *J* is *p* + *v*.
As ‘goodness’ criterion of *v* we take the SSD error function

$$\varepsilon(v) = \sum_{x \in W(p)} (J(x+v) - I(x))^2$$

It measures the image intensity deviation between a neighbourhood of the feature point position in *I* and its potential position in *J* and should be zero in the ideal case. Setting the first derivative of ε(*v*) to zero and approximating *J(x + v)* by its first order Taylor expansion around *v* = 0 results in a better estimate v1. By repeating this multiple times, we obtain an iterative update scheme for *v* which is summarized below [17]:



```
1. Set initial motion vector $v_1 = (0,0)^T$
2. Spatial image gradient $\nabla I = \partial I / \partial(x,y)$
3. Calc. structure matrix $G = \sum_{x \in W(p)} \nabla I(x) \cdot \nabla I(x)^T$
4. for $k = 1$ to $maxIter$
   a) Image difference $\eta(x) = I(x) - J(x + v^k)$
   b) Calc. mismatch vector $b = \sum_{x \in W(p)} \eta(x) \cdot \nabla I(x)$
   c) Calc. updated motion $v_{k+1} = v_k + G^{-1} b$
   d) if $\| v_{k+1} - v_k \| < eps$ then stop (converged)
5. Report final motion vector $v$
```

Table 1: Pseudo-code of the calculation of the motion vector *v* for a given feature point *p*. W(*p*) is a window centered at *p*. Typically the window size is set to 5 x 5 pixel, *maxIter* to 10 and *eps* to 0.03 pixel.

4. Summary of Literature Review

Paper Title	Publications	Authors	Techniques
Visual Object Tracking Based on Local Steering Kernels and Color Histograms[1]	*IEEE Transactions on Circuits & Systems for Video Technology VOL:25 NO:3 YEAR 2013*	Olga Zoidi, Anastasios Tefas, Member, IEEE, and Ioannis Pitas, Fellow	Successfully tracks target object under partial occlusion with slow transformation. Employs appearance based representation method, Color Histogram, LSK tracking technique.
Visual tracking and recognition using appearance: Adaptive models in particle filters[2].	IEEE Trans. Image Process., vol. 13, no. 11, pp. 1434–1456, Nov. 2004.	S. Zhou, R. Chellappa, B. Moghaddam	Appearance based target object representation method is used. Particle Filter Tracking technique is implemented. Single object tracking without occlusion.
Understanding the Basis of the Kalman Filter Via a Simple and Intuitive Derivation[3].	IEEE Signal Processing Magazine SEPT. 2012	Ramsey Faragher	The Kalman filter include smoothing noisy data and providing estimates of parameters of interest.
Tracking video objects in cluttered background[4].	IEEE Trans. Circuits Syst. Video Technol., Apr. 2005.	A. Cavallaro, O. Steiger, and T. Ebrahimi	Hybrid video object tracking, Data association, low level descriptors, object segmentation
Tracking of Multiple Objects under Partial Occlusion[5].	SPIE-Research Gate 2009.	C Bing Han, Christopher Paulson, Taoran Lu, Dapeng Wu, Jian Li.	Template matching, silhouette tracking and trajectory estimation method. Tracks moving object with high accuracy under partial occlusion.
Real-time Visual Tracking of Aircrafts[6].	Digital Image Computing: Techniques and Applications	Ajmal S. Mian.	KLT tracker. Successfully tracks aircraft in motion.

5. Conclusion

This paper presents a literature review of the target object representation methods being grouped into five categories. It also gives a detailed description of the different tracking techniques. This article also reflects that visual object tracking is subjected to varying real life situations like illumination changes, affine transformations and speedy motion of the target object. It also signifies that the existing techniques track object under partial occlusion. Tracking speedy motion of multiple objects simultaneously and under full occlusion is the area that encourages new research.

6. Acknowledgement

I express my sincere gratitude towards my guide Prof. (Mrs.) M. K. Kulkarni for her valuable guidance. I am thankful to Prof. (Dr.) B. D. Phulpagar for having provided me important suggestions and guided me whenever I approached for help. I am also thankful to Prof. (Ms.) D.V. Gore and Prof. (Dr.) S.A. Itkar for their encouragement and support.

References

[1] Visual Object Tracking Based on Local Steering Kernels and Color Histograms Olga Zoidi, Anastasios Tefas, *Member, IEEE, and Ioannis Pitas, Fellow, IEEE* TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY VOL:25 NO:3 YEAR 2013.

[2] Visual tracking and recognition using appearance: Adaptive models in particle filters- S. Zhou, R. Chellappa, and B. Moghaddam IEEE Trans. Image Process., vol. 13, no. 11, pp. 1434–1456, Nov. 2004.

[3] Understanding the Basis of the Kalman Filter Via a Simple and Intuitive Derivation, Ramsey Faragher , IEEE Signal Processing Magazine SEPT. 2012.

[4] A. Cavallaro, O. Steiger, and T. Ebrahimi, “Tracking video objects in cluttered background,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, Apr. 2005.

[5] Tracking of Multiple Objects under Partial Occlusion.C Bing Han, Christopher Paulson, Taoran Lu, Dapeng Wu, Jian Li, SPIE-Research Gate 2009.

[6] Real-time Visual Tracking of Aircrafts. Ajmal S. Mian. Digital Image Computing: Techniques and Applications 978-0-7695-3456-/08 $25.00 © 2008 IEEE DOI 10.1109/DICTA.2008.33.

[7] C. Yang, R. Duraiswami, and L. Davis, “Efficient mean- shift tracking via a new similarity measure,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recogn.*, vol. 1, Jun. 2005, pp. 176–183.

[8] D. Roller, K. Daniilidis, and H. H. Nagel, “Model-based object tracking in monocular image sequences of road traffic scenes,” *Int. J. Comput.Vision*, vol. 10, pp. 257–281, Mar. 1993.

[9] A. Yilmaz, X. Li, and M. Shah, “Contour-based object tracking with occlusion handling in video acquired using mobile cameras,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1531–1536, Nov.2004.

[10] Y. Wang and O. Lee, “Active mesh—a feature seeking and tracking image sequence representation scheme,” *IEEE Trans. Image Process.*, vol. 3, no. 5, pp. 610–624, Sep. 1994.

[11] L. Fan, M. Riihimaki, and I. Kunttu, “A feature-based object tracking approach for realtime image processing on mobile devices,” in *Proc.17th IEEE ICIP*, Sep. 2010, pp. 3921–3924.

[12] L.-Q. Xu and P. Puig, “A hybrid blob- and appearance-based framework for multi-object tracking through complex occlusions,” in *Proc. 2nd Joint IEEE Int. Workshop Visual Surveillance Perform. Evaluation Tracking Surveillance*, Oct. 2005, pp. 73–80.

[13] J. Wang, G. Bebis, and R. Miller, “Robust video-based surveillance by integrating target detection with tracking,” in *Proc. Conf. CVPRW OTCBVS*, Jun. 2006.

[14] N. Papanikolopoulos, P. Khosla, and T. Kanade, “Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision,” *IEEE Trans. Robot. Autom.*, vol. 9, Feb.1993.

[15] G. Welch and G. Bishop, “An introduction to the Kalman filter,” Univ.North Carolina, Chapel Hill, NC, Tech. Rep. TR95041, 2000.

[16] M. Piccardi, “Background subtraction techniques: A review,” in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, vol. 4, Oct. 2004.

[17] Realtime KLT Feature Point Tracking for High Definition Video. Hannes Fassold1 Jakub Rosner2Peter Schallauer1 Werner Bailer1. FAH-2009 GRAVISMA.

[18] Tracking of Multiple Objects under Partial Occlusion.C Bing Han, Christopher Paulson, Taoran Lu, Dapeng Wu, Jian Li, SPIE-Research Gate 2009.

[19] E. Rivlin A. Adam and I. Shimshoni. “Robust fragments-based tracking using the integral histogram,". in Proc. IEEE Conf. CVPR, 2006.

[20] G. R. Bradski. “Computer vision face tracking for use in a perceptual user interface,". in Proc. IEEE Workshop Appl. Comput. Vision, 1998.

[21] T. Okuma. “A natural feature based 3d object tracking method for wearable augmented reality,". in Proc. AMC, 2004.

[22] D. Wu N. Stergiou H. Luo, S. Ci and K. Siu. “A remote markerless human gait tracking for e-healthcare based on content-aware wireless multimedia communications,". IEEE Wireless Commun., vol. 17, no.1 pp. 44-50, 2010