

Target Tracking Using Particle Filter in Least Soft-Threshold Squares Framework

Divya P

*Department of Electronics and Communication,
Amrita University, Coimbatore, India.*

Indhumathi Vivekananda

*Department of Electronics and Communication
Amrita University, Coimbatore, India.*

Abstract

In tracking applications, occlusions and outliers are a major problem faced while tracking in consecutive frames. A Least Soft-threshold Squares (LSS) regression technique that presumes a Gaussian-Laplacian distributed noise is presented. The LSS distance gives the variation between the observation model and dictionary. It effectively handles outliers and provides a precise match to facilitate tracking. The condensation algorithm that uses particle filter for tracking is used to overcome the linearity and non-Gaussian constraints. The particle filter also avoids collision and reduces the dimensionality of the object for further estimations. This technique provides high-quality tracking while cluttering. The LSS regression technique along with particle filter facilitates accurate tracking even in the presence of outliers and occlusion.

Keywords: Occlusions, Outliers, Gaussian-Laplacian, Least Soft-threshold Squares, Particle filter.

I. INTRODUCTION

Visual object tracking is a key area of research that detects, recognizes and tracks objects from a series of images. It locates the target object over time with the help of a source. It has vast applications like visual surveillance, video compression, traffic flow monitoring, and human-computer interaction. It also describes the object

behaviour and replaces the conventional monitoring cameras. Object detection involves locating objects in a video sequence frame. A tracking method involves a detection system in every frame or during the initial appearance of the target. The three main steps involved are video testing, target detection, tracking these targets in every frame, and their analysis for behaviour recognition [1]. Video surveillance uses electro-optical sensors to collect data about the environment. Video cameras are scaled in positions to transmit video streams to the monitoring room. The received video streams are monitored on displays and traced by operators. The operators come across several problems while monitoring. The operator must steer through the cameras, as the target moves between the views of cameras and should not miss any other object while taking it. Consequently monitoring becomes more challenging with a large number of sensors. So, surveillance systems are automated to improve the performance and eliminate the operator errors. The machine vision algorithms are severely affected by occlusions, shadows and weather conditions. Occlusions and outliers interfere our vision during tracking [2]. There are some challenging problems within the surveillance algorithms, such as background modelling, feature extraction, tracking, handling occlusion handling etc.

Sometimes the 3D reconstruction of the images becomes obscure in the presence of occlusion [3]. Tracking algorithms are classified as generative and discriminative. Generative methods rely on the areas that are almost an exact match to the targets tracked. Discriminative methods focus on distinguishing the targets from their backgrounds. The entire tracking mechanism solely depends on these algorithms. The existing algorithms are not efficient in handling occlusions and outliers.

In this paper, a generative tracking algorithm that handles partial occlusion as well as the other challenging factors is presented. An LSS regression technique that assumes the Gaussian-Laplacian distributed noise term. The LSS distance is calculated that gives the variation between an observation sample and the ground truth. It also handles outliers which give an accurate match for object tracking. The LSS regression is used along with particle filter technique that resolves the Gaussian and linearity constraints thereby can be used for a large number of Probability Distribution Functions.

II. ALGORITHMS

A. Least soft-threshold square

The linear regression technique utilizes the linear model to represent a sequence of noisy observations:

$$b = Ya + n \quad (1)$$

where $b \in \mathbb{R}^{m \times 1}$ is an m-dimensional observation vector; $a \in \mathbb{R}^{k \times 1}$ indicates a parameter vector that is k-dimensional; $Y = [y_1; y_2; \dots y_m] \in \mathbb{R}^{m \times k}$ is the input data matrix. Where $n = b - Ia = [n_1; n_2; \dots n_m]$ is the noise term.

The coefficient b is acquired by the maximization of the posterior probability $p(a|b)$, that corresponds to maximizing $p(a, b)$ (joint likelihood). The coefficient b is estimated by $\hat{a} = \arg \max p(b|a) = p(n)$ by the assumption of a uniform prior. This is known as maximum likelihood estimation.

The noise terms $[n_1; n_2; \dots; n_m]$ are independent and identically distributed (i.i.d) as per the PDF $d_\varphi(n_i)$. φ characterises the probability distribution. As a result, the estimator's likelihood is $\prod_{i=1}^m d_\varphi(n_i)$. The objective function

$$L_\varphi(n_1, n_2, \dots, n_m) = \sum_{i=1}^m \sigma_\varphi(n_i) \text{ where, } \sigma_\varphi(n_i) = -\log d_\varphi(n_i).$$

When the random noise $n = b - Ya$ is modelled by the Gaussian distribution ($n_i \in N(0, \sigma_N^2)$), the maximum likelihood estimation is equal to ordinary least squares solution (OLS).

$$\hat{a} = \operatorname{argmin} \frac{1}{2} \|b - Ya\|_2^2 \quad (2)$$

The closed form of solution is $a = (Y^T Y)^{-1} Y^T b$. Though this method is simple to compute, it is susceptible to outliers because of the assumption of Gaussian noise.

If the noise term is modelled by the Laplacian distribution ($n_i \in L(0, \sigma_L^2)$), the maximum likelihood estimation is equal to the least absolute deviations estimation (LAD):

$$\hat{a} = \operatorname{argmin} \frac{1}{2} L \|b - Ya\|_1 \quad (3)$$

The LAD estimation is more robust when compared to the OLS method in terms of handling outliers. But, it is hard to compute using either iterative reweighted least squares methods or the simplex methods [4].

Here, we have modelled the noise term that is linear combination of two components that are independent; an i.i.d Gaussian noise vector ($g_i \in N(0, \sigma_N^2)$) and an i.i.d Laplacian noise vector ($l_i \in L(0, \sigma_L^2)$), hence (1) becomes,

$$b = Ya + g + l \quad (4)$$

where the Gaussian term is used to model the less dense noise terms and the Laplacian term enables the handling of outliers. This is called the Gaussian-Laplacian distribution. Its distribution function $f_{NL}(n_i)$ is given by,

$$f_{NL}(n_i) = f_N(g_i) * f_L(l_i)$$

$$\begin{aligned}
&= \int f_L(l_i) f_N(n_i - l_i) dl_i \\
&= \frac{1}{2\sqrt{2}\sigma_N} \exp\left(-\frac{e_i^2}{2\sigma_N^2}\right) \left[\operatorname{erfcx}\left(\frac{\sigma_N}{\sigma_L} - \frac{e_i}{\sqrt{2}\sigma_N}\right) + \operatorname{erfcx}\left(\frac{\sigma_N}{\sigma_L} + \frac{e_i}{\sqrt{2}\sigma_N}\right) \right] \quad (5)
\end{aligned}$$

where $\operatorname{erfcx}(x) = \exp(x^2) \operatorname{erfc}(x)$

$$\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt$$

The distribution function is very tedious to compute. So we assume the Laplacian term l as a missing term with a value that matches its prior value. Due to this the likelihood estimate is maximised,

$$\begin{aligned}
p(b, a, l) &= p(b|a, l)p(a, l) \\
&= p(b - Ya - l)p(l) \quad (6) \\
&= Z \exp\left\{-\frac{1}{\sigma_N^2} \left(\frac{1}{2} \|b - Ya - l\|_2^2 + \lambda \|l\|_1\right)\right\}
\end{aligned}$$

$$\text{where } Z = \left(\frac{1}{\sqrt{2}\sigma_L}\right)^m \left(\frac{1}{\sqrt{(2\pi)\sigma_N}}\right)^m; \lambda = \frac{\sqrt{2}\sigma_N^2}{\sigma_L}$$

This reduces the value of $\frac{1}{2} \|b - Ya - l\|_2^2 + \lambda \|l\|_1$ with respect to both a and l .

B. The Regression Technique

To maximise the value of (6), take the function:

$$L(x, s) = \frac{1}{2} \|b - Ya - l\|_2^2 + \lambda \|l\|_1, \quad (7)$$

The optimal solution would be $[\hat{a}, \hat{l}] = \arg \min L(a, l)$. To obtain this optimal solution the possible methods are:

- (1) Given \hat{l} , \hat{a} is computed by ordinary least squares method as:

$$\hat{a} = (Y^T Y)^{-1} Y^T (b - \hat{l})$$

- (2) Given \hat{a} , \hat{l} is computed using the shrinkage operation as:

$$\hat{l}_i = S_\lambda([b - Y\hat{a}]_i)$$

where $S_\lambda = \max(|a| - \lambda, 0) \operatorname{sgn}(x)$,

Assume $X=(Y^T Y)^{-1} Y^T$ to start the iteration process. The iteration process comes to an end when a certain criterion based on the requirement is met.

The following are the steps in the regression method:

Input: observation vector b , matrix Y , pre computed matrix X and a constant λ

- i. Initialisation of $l_0=0$ and $i =0$.
- ii. Iteration .
- iii. Compute $a_{i+1} = P(b - S_i)$.
- iv. Compute $l_{i+1} = S_\lambda(b - Y a_{i+1})$.
- v. $i \rightarrow i + 1$.
- vi. Continue the process until termination for the outputs \hat{a} and \hat{l} .

C. Least Soft Square Distance

Let the matrix $Y = [y_1, y_2, \dots, y_k]$, where y_i denotes the i^{th} column of the matrix. After that the vector $Y a$ is computed as $Y a = [y_1 a_1, y_2 a_2, \dots, y_k a_k]$. The matrix Y is known as the dictionary and the vector is called the basis vector. In tracking, the distance between the noisy model and the dictionary contributes to an important aspect in handling outliers; it is inversely proportional to the maximum likelihood,

$$\begin{aligned} \text{dist}(b; Y) &\propto -\log \max p(b, a) \\ &= -\log \max p(b|a)p(a) \end{aligned}$$

In the case of Gaussian-Laplacian distribution, the distance is computed as:

$$\text{dist}_{LSS} = \min \frac{1}{2} \|b - Y a - l\|_2^2 + \lambda \|l\|_1$$

D. Least Soft Square Tracking

In this paper, Hidden Markov Model models track in the Bayesian interface. The state estimation is,

$$\hat{a}_t = \arg \max p(a_t^i | b_{1:t}),$$

a_t^i denotes the i^{th} sample of the state x_t . On the basis of the Bayes theorem $p(a_t^i | b_{1:t})$ is estimated as,

$$p(a_t^i | b_{1:t}) \propto p(b_t | a_t) \int p(a_t | a_{t-1}) p(a_{t-1} | b_{1:t-1}) a_{t-1},$$

The term $p(a_t|a_{t-1})$ indicates the state transition between the two successive frames and $p(b_t|a_t)$ predicts if an object belongs to the class or not. The affine model models the motion and the transition is given by $p(a|a_{t-1}) = N(a; a_{t-1}, \Sigma)$, where Σ indicates a diagonal matrix that denotes the variances of the affine transformation.

Here we perform the Principal Component Analysis (PCA) to generate the target,

$$b = \tau + Uw + g + l$$

where b is the observation vector, w is the basis vector coefficient, U denotes the basis vectors, g is the Gaussian noise term and l is the Laplacian noise term.

The LSS distance is calculated by;

$$dist(b; U, \tau) = \min \frac{1}{2} \|\bar{b} - Uw - l\|_2^2 + \lambda \|l\|_1$$

$\bar{b} = b - \tau$, The likelihood estimation is given by:

$$[\hat{z}^i, \hat{l}^i] = \arg \min \frac{1}{2} \|\bar{b}^i - Uw^i - l^i\|_2^2 + \lambda \|l^i\|_1$$

i denotes the i^{th} model of the state a .

The observation likelihood is estimated as:

$$p(b^i|a^i) = \exp(-\gamma dist(b; U, \tau)),$$

γ is the Gaussian kernel shape.

The non zero value of the Laplacian term predicts the presence of outliers [4]. The target state in each frame is obtained and the value of the observation vector is extracted. l_0 is estimated and from this value, the observation vector is reconstructed by the replacement of the Laplacian values with τ .

E. Particle Filter

Under the assumption of non linear state and non Gaussian noise, the Particle filtering method, also known as sequential Monte Carlo methods are particularly appealing. It primarily involves a set of weighted particles observed in accordance with the Monte Carlo incorporation principles [5]. It can be applied under general hypotheses, easy to implement and is capable of out ruling heavy clutter.

Consider a prior probability $p(M_1)$, a transition prior probability $p(M_T|M_{T-1})$ along with a Likelihood $P(N_T|M_T)$, the particle filter algorithm is as follows:

Step 1- Initialization

Initialize, $T = 1$. For $j = 1, \dots, N$, and sample $(M_1^j) \sim p(M_1)$ and increment $T = 2$.

Step 2- Importance Sampling step

For $j = 1, \dots, n$ and sample $Mp_T^{(j)} \sim p(Mp_T^{(j)} | M_{T-1}^{(j)})$ moreover set $Mp_{1:T}^{(j)} = (M_T^{(j)}, M_{1:T-1}^{(j)})$. For $j = 1 \dots, n$, estimate the importance weights. $W_T \sim p(N_T | M_T^{(j)})$ and normalize the importance weights.

Step 3- Selection Step

Resample with replaced N particles $[M_{j:T}^{(j)}, j = 1, \dots, n]$ from $[Mp_{1:T}(j), j = 1, \dots, n]$ in accordance with the normalized importance weights.

Increase $T = T + 1$ and go to Step 2.

A primary particle filtering algorithm includes two main processes, sampling and selection [5]. In sampling, the individual weights are sampled for a particular time t , and the importance weights are normalised. In the selection process, particles with high or low importance weights are multiplied or discarded correspondingly to attain the required number of particles. This enables efficient tracking of moving objects.

III. EXPERIMENT

The proposed algorithm is executed in MATLAB which runs at 4 frames/second on a PC with Intel i5-4200M 2.5 GHz and 8 GB memory. λ (regularization constant) is assumed to be equal to 0.1. For each sequence, the target location is tagged manually in the initial frame. The images are resized to 32*32 pixels. 16 eigenvectors are used for the PCA representation. 600 pixels are sampled at a time at batches of 5 frames. This algorithm is applied to the Caviar dataset. The proposed method utilizes the techniques of particle filter and least soft-threshold squares. The process of tracking starts with target identification. After the target is identified, the particle filter converts a 3-dimensional real life object into a 2-dimensional image. This facilitates the further processing as this reduces the number of estimation parameters. Particle filtering technique categorises pixels of same characteristics and samples them which reduces the overall time taken for tracking. The LSS follows the Gaussian-Laplacian distribution which eliminates occlusions and outliers. The difference between the ground truth and the present frame is calculated. The value corresponds to the

coordinate values of pixels that vary with respect to the ground truth. This gives the LSS distance.

IV. RESULTS

The proposed algorithm efficiently tracks the target in the presence of outliers and occlusions [a] and [b] and the following parameters are estimated.

LSS Distance

The value of the LSS distance that effectively handles outliers is computed for a constant value of lambda. The value of lambda is assumed to be either 0.05 or 0.1 in general. We assume it to be 0.1.

For the assumed lambda value, the LSS distance calculated is:

Table 1. LSS Distance

Value of lambda	0.1
LSS distance (w.r.t target)	10.466
LSS distance (w.r.t outlier)	12.177

The LSS distance with respect to the target is less than the distance computed with respect to the outlier. Hence in the presence of an outlier the algorithm continues to track the target as it opts for the lesser value.

Overlap Rate and Center error evaluation

The values of the overlap rate and the center error on the basis of the target region are computed and are compared to that of the existing techniques [4]. The proposed method has a much lower value than those of the existing ones.

Table 2. Overlap Rate and Center error evaluation

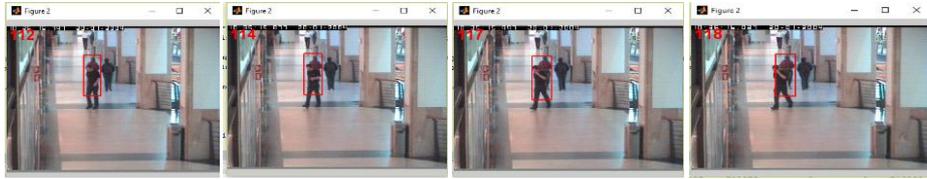
PARAMETER	LOT	SCM	SCC	LSS
OVERLAP RATE	11.78	0.811	0.752	0.1262
CENTER ERROR	0.599	2.53	1.72	1.2

V. CONCLUSION

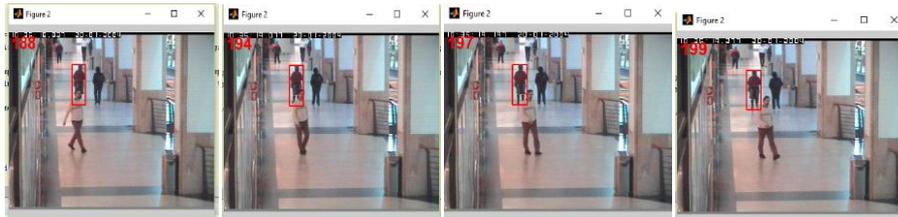
In this paper, the algorithm initially updates the appearance model and then uses the information of the received frame to check if the model updation is valid or not. This avoids the improbability in the validation model. The current appearance model to corroborate the predicted target appearance features is used in the existing algorithms. The goal of updating the model is to include new alterations in variations of the appearance to the model updated, while the existing techniques do not have stipulation to include transformations in model updating through authorization changes with respect to the existing model. Experimental outcomes on the video sequence prove that the algorithm performs better than the existing ones in terms of outliers, occlusions and appearance variation co-existing scenarios. This method can also be used for various other applications such as facial recognition, air borne radar tracking etc.

REFERENCE

- [1] Yilmaz, A., Javed, O. and Shah, M. (2006). Object tracking. CSUR, 38(4), p.13-es.
- [2] Praveena, J, 2013. A Hybrid Method for Occlusion Removal. International Journal of Application or Innovation in Engineering & Management, [Online]. 2, 111-118. Available at:<http://www.ijaiem.org/volume2issue7/IJAIEM-2013-07-11-029.pdf> [Accessed 22 March 2016].
- [3] Antony M, B. (2012). Removing Occlusion in Images using Sparse Processing and Texture Synthesis. IJCSEA, 2(3), pp.117-124.
- [4] Wang, D., Lu, H. and Yang, M. (2015). Robust Visual Tracking via Least Soft-threshold Squares. IEEE Trans. Circuits Syst. Video Technol., pp.1-1.
- [5] Hue, C., Le Cadre, J. and Perez, P. (2002). Tracking multiple objects with particle filtering. IEEE Trans. Aerosp. Electron. Syst., 38(3), pp.791-812.
- [6] Islam, Md , 2009. Video Based Moving Object Tracking by Particle Filter. International Journal of Signal Processing, Image Processing and Pattern, [Online]. 2, 119-132. Available at: http://www.sersc.org/journals/IJSIP/vol2_no1/11.pdf [Accessed 21 March 2016].
- [7] Yuan, Y., Yang, H., Fang, Y. and Lin, W. (2015) ‘Visual object tracking by structure complexity coefficients’, IEEE Transactions on Multimedia, 17(8), pp. 1125–1136. doi: 10.1109/tmm.2015.2440996.
- [8] CAVIAR: Context Aware Vision using Image-based Active Recognition. 2016.
- [9] CAVIAR:Context Aware Vision using Image-based Active Recognition. [ONLINE] Available at: <http://homepages.inf.ed.ac.uk/rbf/CAVIAR/>. [Accessed 05 October 2016].



[a] Tracking using Caviar dataset under occlusion



[b] Tracking using Caviar dataset in the presence of outliers