

Fast Visual Object Tracking Using Modified kalman and Particle Filtering Algorithms in the Presence of Occlusions

G.Mallikarjuna Rao

Scientist 'E', DRDO (RCI), Hyderabad, Andhra Pradesh, India
Email:mallikarjun.guttikonda@gmail.com

Siva Prasad Nandyala

Research Scholar, Department of Electronics and Communication Engineering, NIT Warangal,
Warangal, Andhra Pradesh, 506004, India
Email:speech.nitw@gmail.com

Dr.Ch.Satyanarayana

Professor, Department of Computer Science and Engineering,
Jawaharlal Nehru Technological University Kakinada (JNTUK). A. P., India
Email:chsatanarayana@yahoo.com

Abstract—In the present day real time applications of visual object tracking in surveillance, it has become extremely complex, time consuming and tricky to do the tracking when there are occlusions are present for small duration or for longer time and also when it is done in outdoor environments. In these conditions, the target to be tracked can be lost for few seconds and that should be tracked as soon as possible. As from the literature it is observed that particle filter can be able to track the target robustly in different kinds of background conditions, and it's robust to partial occlusion. However, this tracking cannot recover from large proportion of occlusion and complete occlusion, to avoid this condition, we proposed two new algorithms (modified kalman and modified particle filter) for fast tracking of objects in the presence of occlusions. We considered the complete occlusion of tracking object and the main objective is how fast the system is able to track the object after the occlusion is crossed. From the experimental results, it is observed that the proposed algorithms have shown good improvement in results compared to the traditional methods.

Index Terms—Visual object tracking, kalman filter, background subtraction, particle filter.

I. INTRODUCTION

Object tracking is a fundamental and an important problem of dynamically extracting two-dimensional (2D) information in most visual applications including image processing, computer vision, video surveillance, human computer interaction (HCI)[1]. From the last few years, it has become a substantial research in the field of computer

vision, more specifically, in the tracking of moving objects when the occlusions are present for longer time. As the core basis of military systems, security observance, robot vision and intelligent security system, it's been one in every of the recent spots in the field of computer vision analysis. As a result of tracking usually uses a single, position fixed camera, and the high-speed moving target brings problems concerning dynamic interaction, Small target size, background clutter, low contrast with the background, appearance changes ,erratic motion and occlusions are some of the most common problems that make reliable tracking difficult to achieve.

From the above mentioned problems, the main challenge is the difficultness of target tracking formula. Among of these issues, the occlusion brings huge troubles. Therefore, a way to handle the occlusion is that the burning problem in an exceedingly robust tracking system [2], [3].

A filter is an algorithm that provides an efficient computational means to estimate the state of a dynamic system from a series of measurements. Filtering techniques can be used to improve model predictions by updating the model state variables sequentially i.e. each time an observation is available [4].

This paper is organized as follows, in section II explains about the basics of kalman and particle filter, Section III explains about the Visual object target tracking process using kalman and particle filter. Section IV explains about the proposed modified kalman and particle filter for the visual object target tracking, Section V gives the experimental results for visual object tracking for the proposed methods and conclusions are given in section VI.

II. KALMAN FILTER

The Kalman filter could be an algorithmic answer to the discrete-data linear filtering problem. There has been a lot of analysis has been done on this specific filter and it has been used extensively within the field of visual object tracking. As from the literature, the Kalman filter is one in every of the foremost widespread technique that's used for visual object tracking [5], [6].

A. The basic idea

The Kalman filter offers an estimate of the state of a dynamic system from noisy measurements. It offers a recursive minimum variance estimate of the state of the system. To be ready to apply the Kalman filter to present estimate of the position in setting, a discrete-time state space model is required. This consists of a state equation and a measuring equation.

The state equation is given by:

$$x_k = A_k x_{k-1} + B_k u_k + \varepsilon_k \quad (1)$$

Explanation of the different variables and symbols are as follows:

1. x_k is an n-dimensional state vector , which is to be estimated
2. A_k is an n-dimensional known system matrix
3. B_k is an n by m-dimensional known input matrix
4. u_k is an optional m-dimensional input vector
5. ε_k represents the process noise
6. y_k is a p-dimensional measurement vector
7. C_k is a p by n-dimensional known measurement matrix
8. δ_k represents the measurement error

Because variable u_k is optional, this u_k can be left out of the equation. So it gives a new state equation which can be simplified into:

$$x_k = A_k x_{k-1} + \varepsilon_k \quad (2)$$

The measurement equation is:

$$y_k = C_k x_k + \delta_k \quad (3)$$

The variables ε_k , δ_k are assumed to have zero cross correlation and are white and ε_k and δ_k both of them

have a covariance matrix (Q_k and R_k respectively) and have normal probability distributions which are

$$\begin{aligned} p(\varepsilon_k) &\sim N(0, Q_k) \\ p(\delta_k) &\sim N(0, R_k) \end{aligned} \quad (4)$$

From now onwards the system matrix A_k and measurement matrix C_k will remain constant. But, it is possible chance that both matrices could change with each measurement or time-step, covariance matrices Q_k and R_k are assumed to remain constant.

Prediction equations:

The problem of estimating the state of a system is given as follows, With Kalman filters, and the initial state is understood. The initial state is delineating by the state

vector x_0 . To be ready to predict the state of the system at

time k+1, the previous estimate \hat{x}_k is required. as a result of the prediction makes use of the recent estimate, before

incorporating the new measurements found at time k_1 , this estimate named as ‘‘a priori’’ estimate and this priori

estimate is specified by \hat{x}_{k+1}^- , here the hat denotes that it's estimate and therefore the higher minus denotes that it's a previous estimate. Therefore to calculate the a priori estimate, the subsequent equation is often formed:

$$\hat{x}_{k+1}^- = A \hat{x}_k \quad (5)$$

In other words, to obtain a new estimate, which will multiply the last collected estimate \hat{x}_k with A, which is the system matrix and also needed to change the error covariance matrix P_k^- , which is associated with the a priori estimate with every new time step.

The below equation shows on how to calculate a new value for P_k^- :

$$P_{k+1}^- = A P_k A^T + Q \quad (6)$$

It looks that counting on the dimensions of Q, the random process noise variance matrix features an important result on the error variance matrix. The worth of Q is often obtained before victimization it within the method that's to be calculable. But this tends to be somewhat of a tough task Q needs to simulate method noise for the method to be calculable, however it's difficult to get the amount of noise before the method is running. Unremarkably acceptable results may be reached once enough uncertainty is infused into the method by

giving Q an inexpensive price. but the Kalman filter are able to do higher results once appropriate values for Q square measure found through standardization it with the assistance of a definite filter, this method is named system identification.

Update equations:

The equations required for changing the state of the system square measure given below as can be seen within the previous section, we tend to currently have associate a priori estimate \hat{x}_k^- . What's currently required is associate updated estimate \hat{x}_k , this is often known as the a posteriori estimate. to get the a posteriori estimate, the strident measurement y_k should be combined with the a priori estimate. This is often shown within the following equation:

$$\hat{x}_k = \hat{x}_k^- + K_k (y_k - C\hat{x}_k^-) \tag{7}$$

From the above "(7)", the a posteriori estimate \hat{x}_k can be obtained by combining the a priori estimate \hat{x}_k^- with the weighted difference between the actual measurement y_k and the measurement prediction $C\hat{x}_k^-$. The difference $(y_k - C\hat{x}_k^-)$ is called the residual. This reflects the discrepancy between the actual measurement y_k and the measurement that was predicted. It appears that the prediction of the value of the next measurement and the actual measurement are the same if the residual is zero.

From the "(7)" there still is one symbol that is still to be explained. The Kalman gain K_k is a matrix which aims to minimize the a posteriori error covariance matrix. The equation for calculating the Kalman gain is:

$$K_k = P_k^- C^T (C P_k^- C^T + R)^{-1} \tag{8}$$

From the above "(7)", it is observed that when the measurement error covariance matrix R reaches zero, which implies the Kalman gain will weigh the residual more heavily:

$$\lim_{R_k \rightarrow 0} K_k = C^{-1} \tag{9}$$

In this condition, the actual measurement y_k is trusted more and more and also at the same time, the predicted measurement $C\hat{x}_k^-$ is trusted less and less.

If however the a priori estimate error covariance matrix P_k^- reaches zero, it indicates the Kalman gain will weigh the residual less heavily:

$$\lim_{P_k^- \rightarrow 0} K_k = 0 \tag{10}$$

If this is the situation and covariance matrix P_k^- approaches zero, the actual measurement is trusted less and less, on the other hand $C\hat{x}_k^-$ is trusted more and more.

The error covariance update equation is given as follows

$$P_k = (I - K_k C) P_k^- \tag{11}$$

The error covariance matrix P_k which gives a statistical measure of the uncertainty in x_k , it is a measure of the dispersion of x_k around \hat{x}_k .

The Kalman filter estimates a definite method by employing a type of feedback management. Initially it estimates the state of the system at a given time and afterward receives feedback from incoming measurements. In alternative words, initial it predicts what the state of the system are and, once receiving new measurements from sensors, these predictions are corrected.

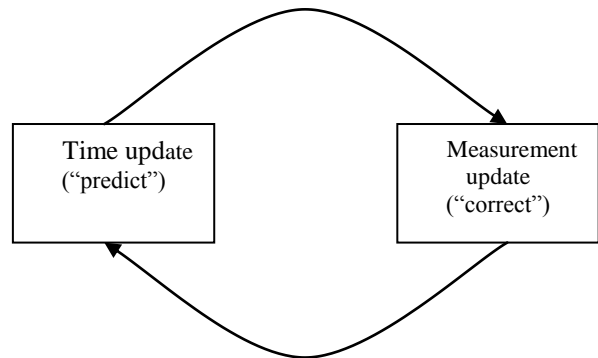


Fig. 1: The representation of Kalman filter cycle

From the above Fig. 1, the Kalman filter is an ongoing process. For the first time the Kalman filter is used and it makes use of the initial values given at time t_0 . It is assumed that system matrix A , measurement matrix C , and noise covariance matrices Q and R are constant all the time.

The first step to be taken now's to use the time update equations. These are answerable for projecting these values of the state and error variance estimates forward in time. Once this step is completed, there'll be new a priori estimates accessible for a next time step. To do this, getting a replacement system state estimate and a replacement error variance estimate matrix, the previous state and error variance estimates are used. If the method has simply started the initial values are going to be used, otherwise the last a posteriori estimates are going to be

accustomed calculate the new a priori estimates. How it is

After getting new a priori estimates, consequent step is to use the measurement update equations. As tacit by the name of those equations, they create use of feedback, within the type of new measurements, and also the a priori estimates to calculate improved a posteriori estimates. First the Kalman gain needs to be calculated, to envision what quantity the residual are going to be weighed. Once the Kalman gain is thought, it's potential

done can be seen in Fig. 2.

to get a replacement a posteriori state estimate. As it is shown in figure 2, which is able to calculate the new state, the new measurement is compared to foreseen measurement, increased with the Kalman gain and combined with the a priori state estimate. The ultimate step is getting associate degree a posteriori error variance estimate. All the measurement update equations square measure conferred in Fig. 2

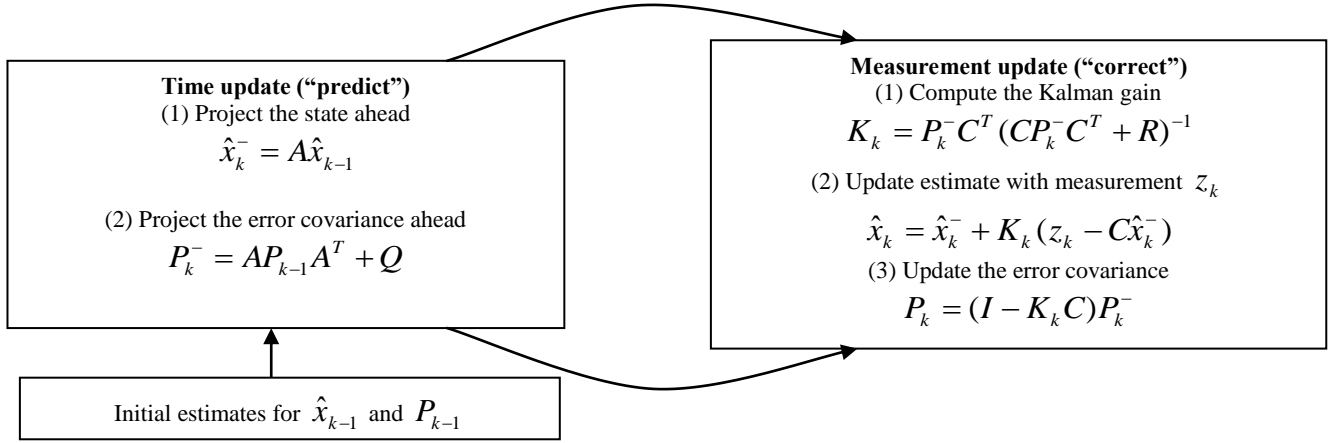


Fig. 2: The Kalman filter algorithm

The Kalman filter incorporates a algorithmic nature, therefore this whole method are perennial till program termination. In each cycle of the method, i.e. once whenever and measuring update combine, the previous a posteriori estimates square measure accustomed calculate the new a priori estimates and from them the new a posteriori estimates.

PROPOSED ALGORITHM

Modified kalman filter

Initial estimates for \hat{x}_{k-1} and P_{k-1}
Initialize Q and R matrices

1. Time update (“predict”)

(i) Project the state ahead

$$\hat{x}_k^- = A \hat{x}_{k-1} \quad (12)$$

(ii) Project the error covariance ahead

$$P_k^- = A P_{k-1} A^T + Q \quad (13)$$

2. Measurement update (“correct”)

(i) Compute the Kalman gain

$$K_k = P_k^- C^T (C P_k^- C^T + R)^{-1} \quad (14)$$

(ii) Update estimate with measurement z_k

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - C \hat{x}_k^-) \quad (15)$$

3. If occlusion occurs, i.e. measured state is not available

Then propagate the system states with dynamically calculated state estimation technique until measured state is available

$$\hat{X}_k = \hat{X}_k^- + \frac{(\hat{X}_k^- - \hat{X}_1)}{k} * \Delta k \quad (16)$$

Where Δk is the state update interval time

4. Update the error covariance

$$P_k = (I - K_k C) P_k^- \quad (17)$$

Repeat the steps 1 to 4 until tracking completes.

III. PARTICLE FILTER

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. By using particle filtering this limitation can be over taken.[7].

The fundamental and basic plan of a particle filter is to represent the posterior density of estimates making the point masses or particles and procure the estimates from

this density. With every measurement, the particles area unit updated in order that they represent the new posterior resulting in the new estimates. Thus, one has got to sample the posterior density operate to get the particles. However, within the estimation method, posterior density isn't sometimes acknowledged. Hence, associate degree approximation to the posterior, referred to as importance operate or proposal operate that spans the complete calculable parameter area is employed to get samples. The particle filter algorithmic program provides a mechanism to consecutively sample the importance operates and update the samples to represent the posterior. So as to make sure that the samples effectively span the complete parameter area, a resampling procedure is additionally performed at every cycle [8, 9].

Particle filtering primarily combines the particles at a specific position into one particle, giving that particle a weight to replicate the quantity of particles that were combined to create it. This eliminates the necessity to perform redundant computations while not skewing the likelihood distribution. This can be Particle filtering accomplished by sampling the system to make N particles, then scrutiny the samples with each other to get an importance weight. once normalizing the weights, it resample N particles from the system utilizing these weights. This method greatly reduces the quantity of particles that has to be sampled, creating the system a lot of less computationally intensive [10].

Particle Filters use multiple distinct “particles” (samples) to represent the distribution over the position of a tracked target. Every object is tracked exploitation of multiple particles. The Particles have a set of parameters that square measure wont to outline the state of our moving target in an exceedingly very uproarious background. Counting on the importance function and therefore the resampling approaches, there are many different types of particle filters. Given N particles (samples) at a time t-1, close to distributed per the posterior distribution, the particle filters will enable us to compute N particles at time t.

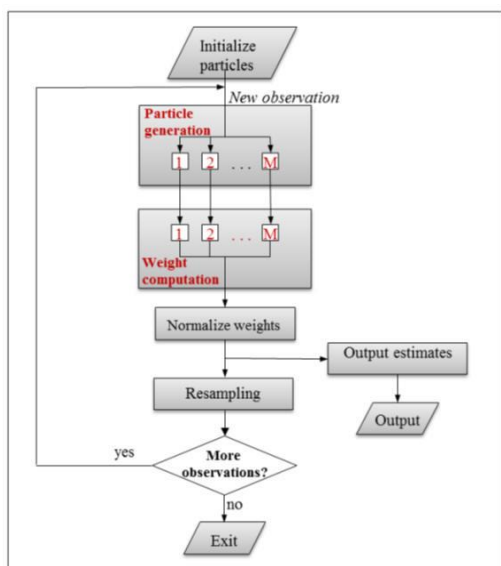


Fig. 3: Particle filter algorithm [11]

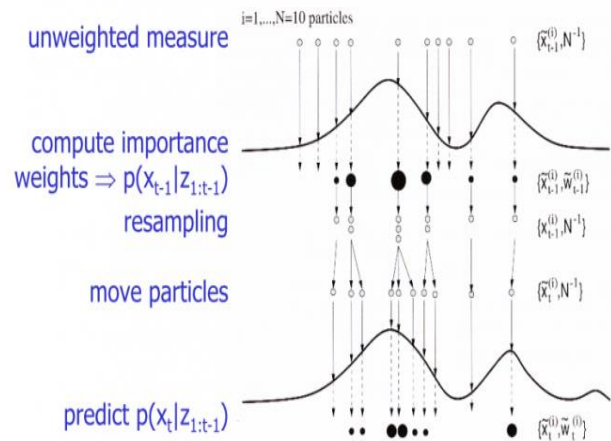


Fig 4: Particle filter algorithm visualization [12]

The basic algorithm of particle filter is described in the following simple steps:

- 1) Initialize the filter by guessing N points in the environment (assuming a uniform distribution); this is the set of N particles.
- 2) Based on the next control input(s), estimate new positions for the particles.
- 3) Based on the next observation, determine the probability of each particle.
- 4) Normalize the probabilities so that they sum to 1.
- 5) Randomly sample with replacement a new set of M particles from the set of N particles, with the probability of a particle being chosen equal to the normalized probability computed in step 4.
- 6) Keep the new set of particles and throw out the old set. Go to 2.

The summary of the particle filter is outlined in the following steps [13],[14].

The system as well as measurement equations are given as follows:

$$\begin{aligned}
 X_{k+1} &= f_k(X_{k+1}, w_k) \\
 y_k &= h_k(X_k, v_k)
 \end{aligned}
 \tag{18}$$

Where k is the time index, x_k is the state, w_k is the process noise, y_k is the measurement, and v_k is the measurement noise. The functions $f(\cdot)$ and $h(\cdot)$ are time-varying nonlinear system and measurement equations. The noise sequences $\{w_k\}$ and $\{v_k\}$ are assumed to be independent and white with known pdf's. y_k is an observation related to x_k by a function h_k .

1. Assuming that the pdf of the initial state $p(x_0)$ is known, randomly generate N initial particles on the basis of the pdf $p(x_0)$. These particles are denoted as $X_{0,i}^+ (i = 1, \dots, N)$. The parameter N is chosen by the user as a tradeoff between computational effort and estimation accuracy

2. For $k = 1, 2, \dots$, do the following

- a) Execute the time propagation step to get a priori particles $X_{k,i}^-$ using the known process equation and the known pdf of the process noise:

$$X_{k,i}^- = f_{k-1}(X_{k-1,i}^+, W_{k-1}^i) (i = 1, \dots, N) \quad (19)$$

Where each W_{k-1}^i noise vector is randomly generated on the basis of the known pdf of w_{k-1} .

- b) Calculate the relative likelihood q_i to each one particle $X_{k,i}^-$ and conditioned on the measurement y_k . This can be done by evaluating the pdf $P(Y_k / X_{k,i}^-)$ on the basis of the nonlinear measurement equation and the pdf of the measurement noise.
- c) Surmount the relative likelihoods which are obtained from the previous step as follows:

$$q_i = \frac{q_i}{\sum_{j=1}^N q_j} \quad (20)$$

The summation of all the likelihoods which is equal to one.

- d) Generate a set of a posteriori particles $X_{k,i}^+$ on the basis of the relative likelihoods q_i . This step is called the resampling step.
- e) Now that we have a set of particles $X_{k,i}^+$ that are distributed according to the pdf $P(X_k / y_k)$, we can compute any desired statistical measure of this pdf. The most interested part is computing the mean and the covariance.

IV. PROPOSED ALGORITHM

Modified Particle Filter:

From the above given particle filter description, it's discovered that the tracking of object isn't quick once the occlusion is for extended time, to avoid this, we are proposing the subsequent modifications to the particle filter during which the following proposed algorithm gives better performance. Presume that throughout the initialization of the particle filter at time $t=0$, we have sampled the prior $p(x_0)$ and obtained a random sample set $X_0 = \{x_0^i\}$, with $i=1, \dots, I$.

In the prediction part, we tend to use the set of samples X_t containing the particles from the last iteration of the particle filter method, or, if it is the first time the algorithm is run, a group of particles which are randomly distributed X_0 . From X_t all of the particles are used, and also the transition model which is $p(x_{t+1} / x_t^i)$ used to sample from. As this transition results in a new set of particles X_{t+1}' , the prime denotes the fact that in the obtained set, the new measurement has not incorporated yet. The new set estimates the prior density $p(x_{t+1})$.

y_{t+1} is integrated into the process of calculating the posterior density in measurement update phase. Since the prior density is available, it is approximated with the set of particles X_{t+1}' . Every particle has a certain weight which is linked to it. This weighing function is given by:

$$\Pi_{t+1}^i = p(y_{t+1} / x_{t+1}^i) \quad (21)$$

All of the weights of the particles are given according to a given observation likelihood. The result is a set of weights, which are not normalized yet. However this will be done by the end of the update phase.

The new set of particles X_{t+1} is obtained to be resampling from the weighted set $\{x_{t+1}^{ii}, \Pi_{t+1}^i\}$.

During resampling phase, particles which are having a high weight are more probably to be drawn from the weighted set, some particles will most probably be picked several times. In general with a lower valued weight, have a more chance of being discarded. After getting the new set of particles, the weights yet to be normalized. This will be done in the last normalization-phase, which guarantees that sum of all weights will lead to one.

We have a uniformly distributed set of samples X_0 , obtained from sampling from the prior $p(x_0)$

Step 1) for each particle x_t^i :

- get a sample x_{t+1}^{ii} from the transition model $p(x_{t+1} / x_t^i)$
- a new set of samples X_{t+1}' is obtained

Step 2) for $i=1 \dots I$

- the weight of each particle x_{t+1}^{ii} from x_{t+1}' by the likelihood $\Pi_{t+1}^i = p(y_{t+1} / x_{t+1}^{ii})$ for $j=1 \dots I$
- Draw one x_{t+1} sample x_{t+1}^j from $\{x_{t+1}^{ii}, \Pi_{t+1}^i\}$

- normalize all weights \prod_{t+1}^i

Step 3) If occlusion occurs, i.e measured state is not available

Then propagate the system states with dynamically calculated state estimation technique until measured state is available

$$X_{t+1} = X_t + \frac{(X_t - X_1)}{t} * \Delta t \tag{22}$$

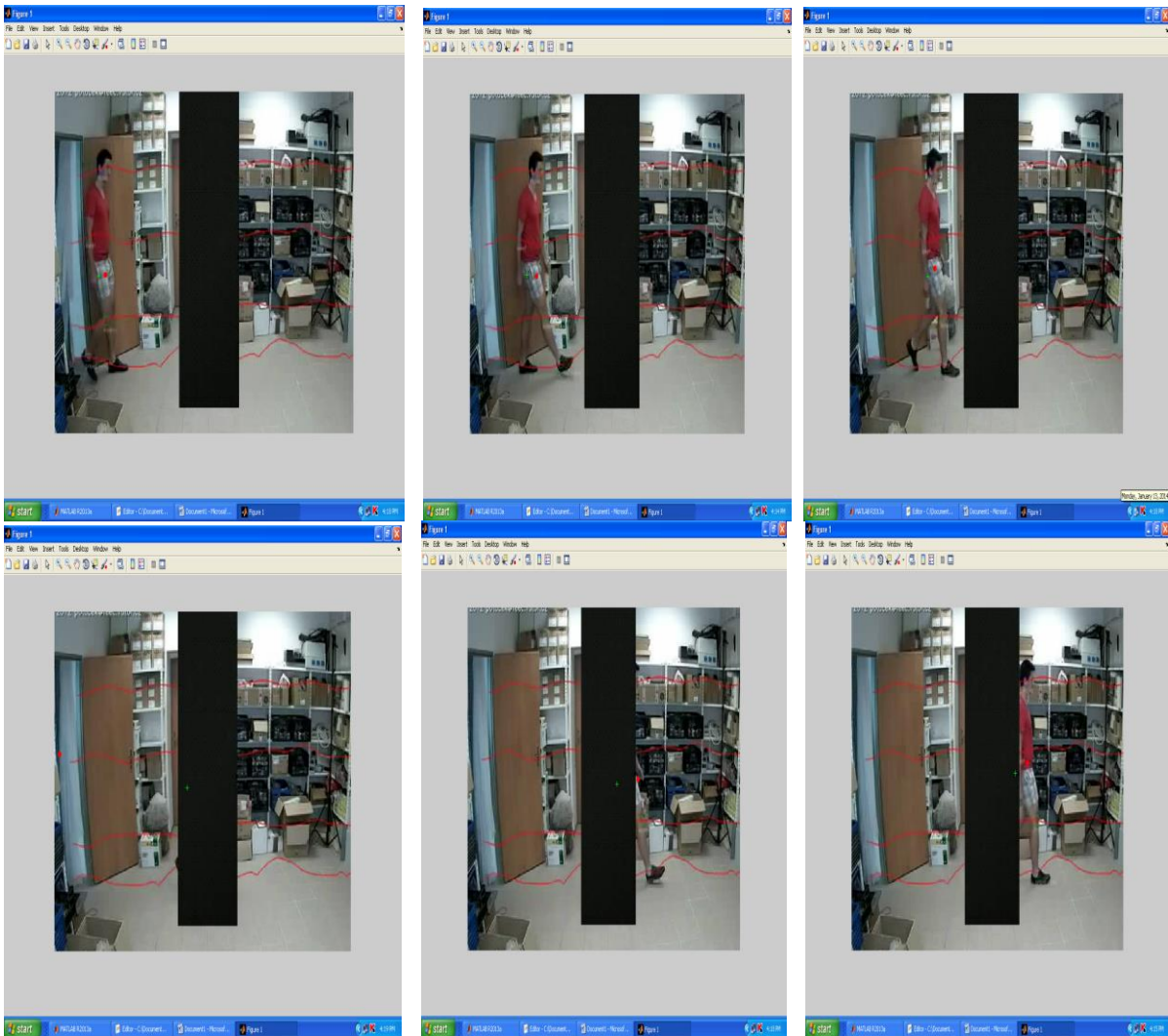
Where Δt is the state update interval time

Obtain the measurement state y_{t+1} from the updated system states X_{t+1}

- A new set of samples X_{t+1} is obtained. It approximates the posterior density $p(x_{t+1} / y_{t+1})$.

Step 1, 2 and 3 are repeated for the duration of the process tracking.

V. RESULTS AND DISCUSSIONS



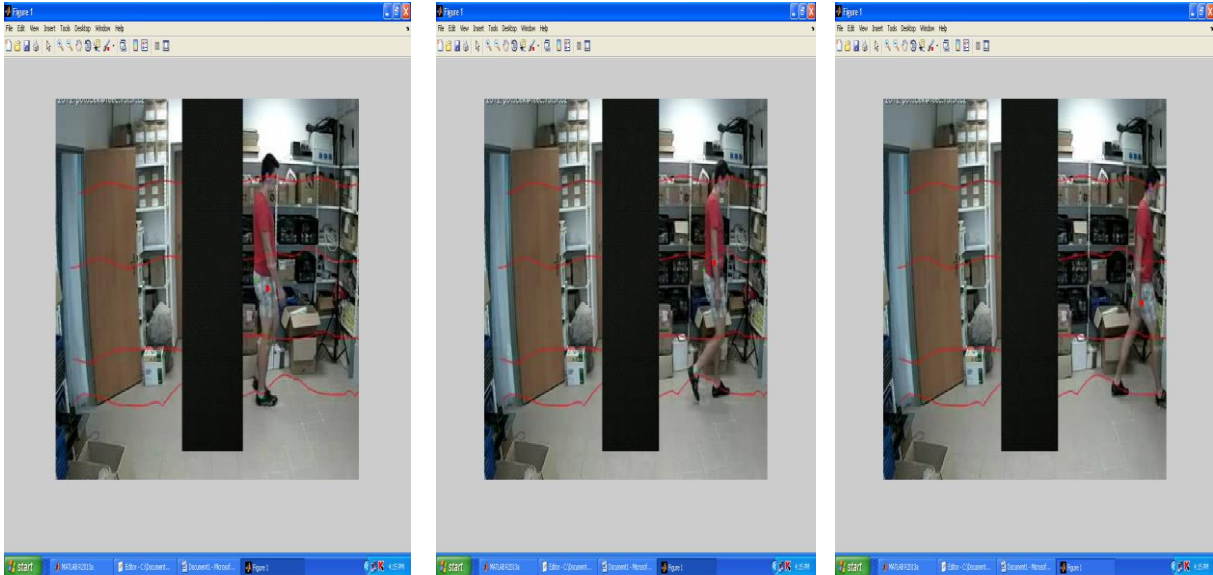


Fig. 5: Visual Object tracking using the modified Kalman Filtering

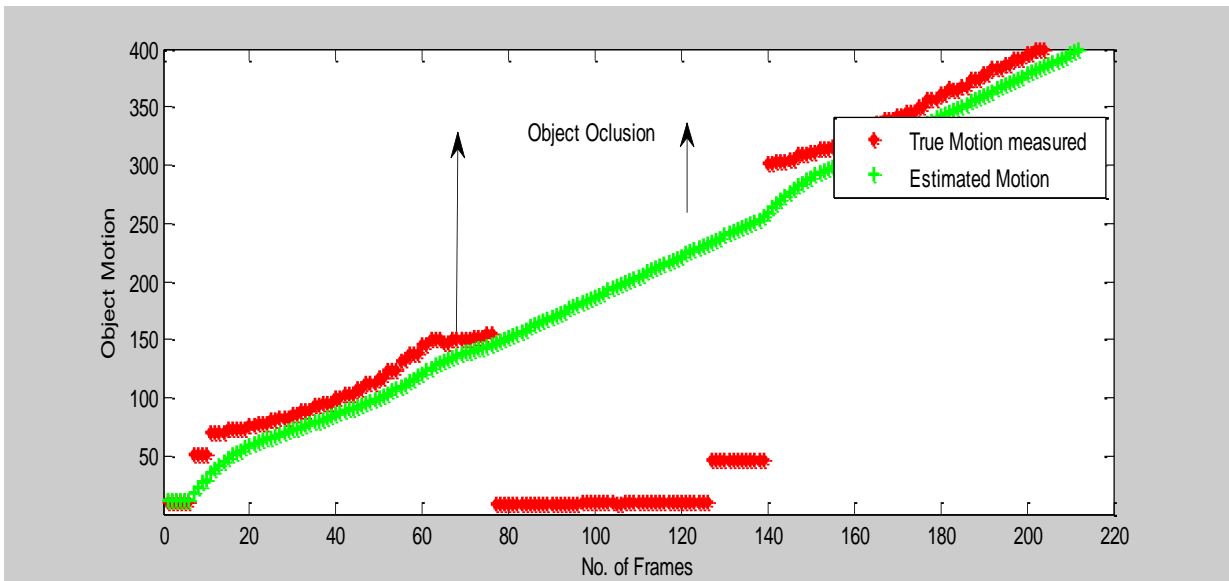
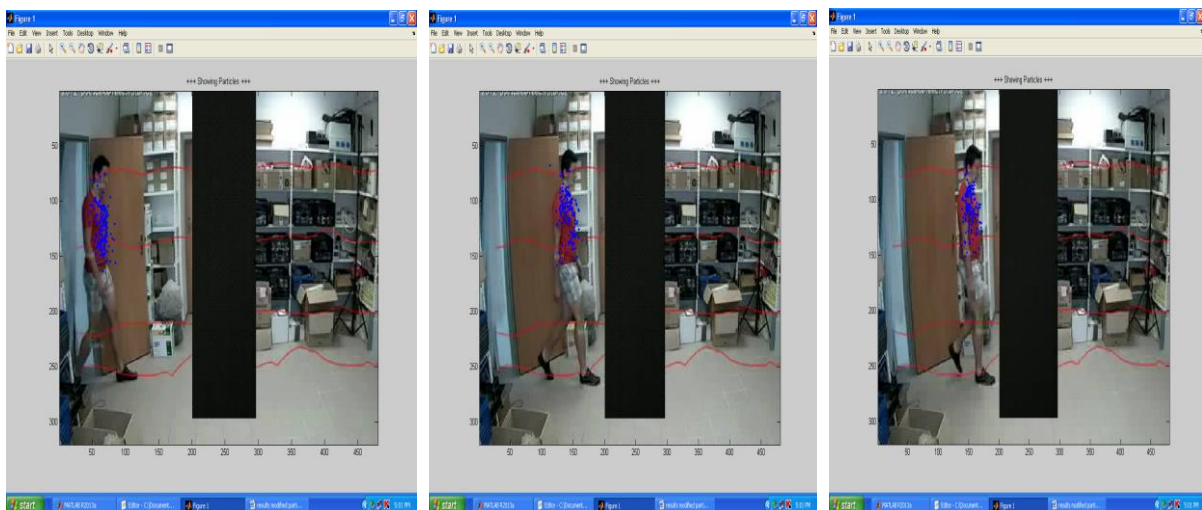


Fig.6: The Tracking Performance using the modified Kalman Filter



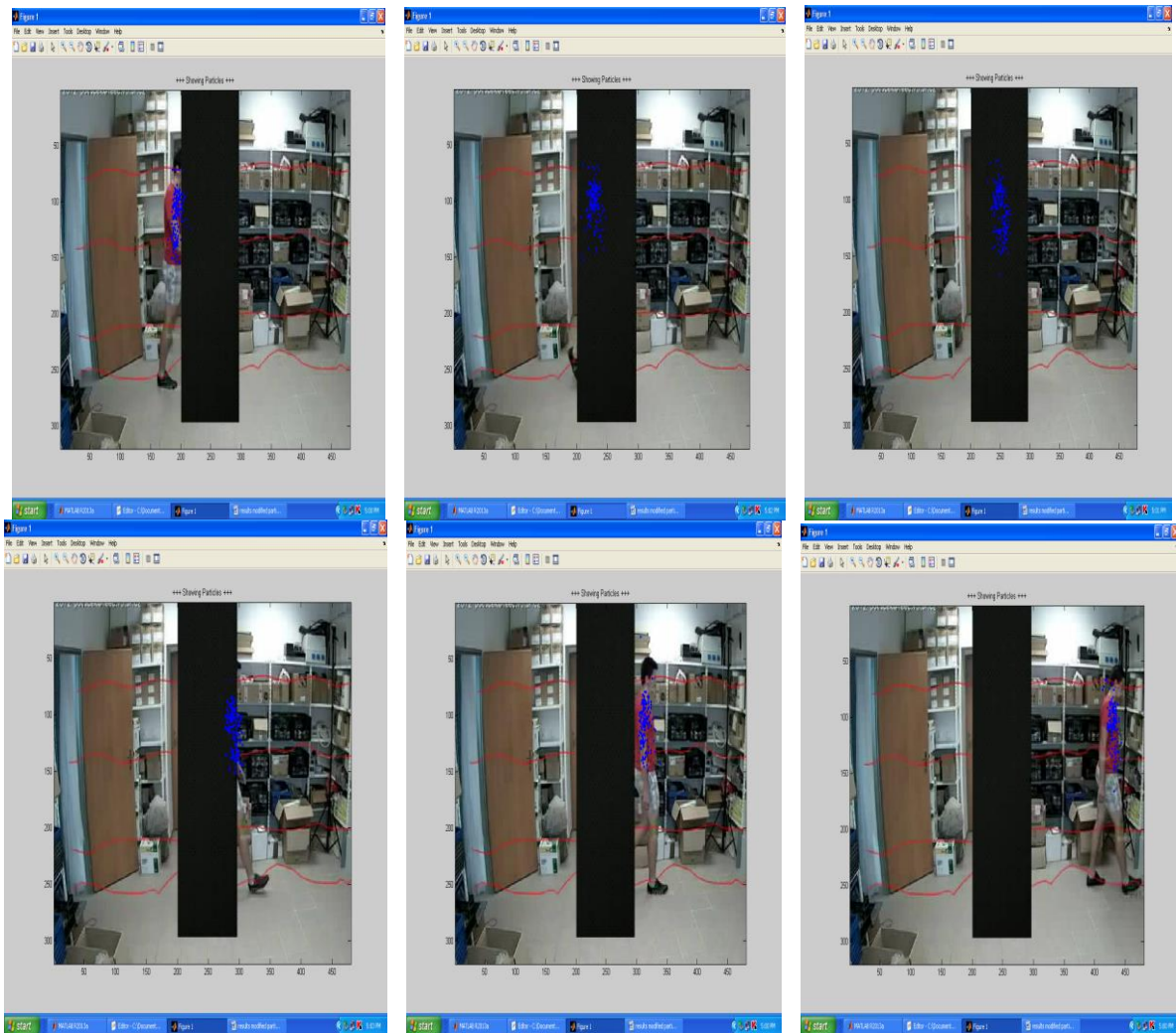


Fig. 7: Visual Object tracking using the modified Particle Filtering

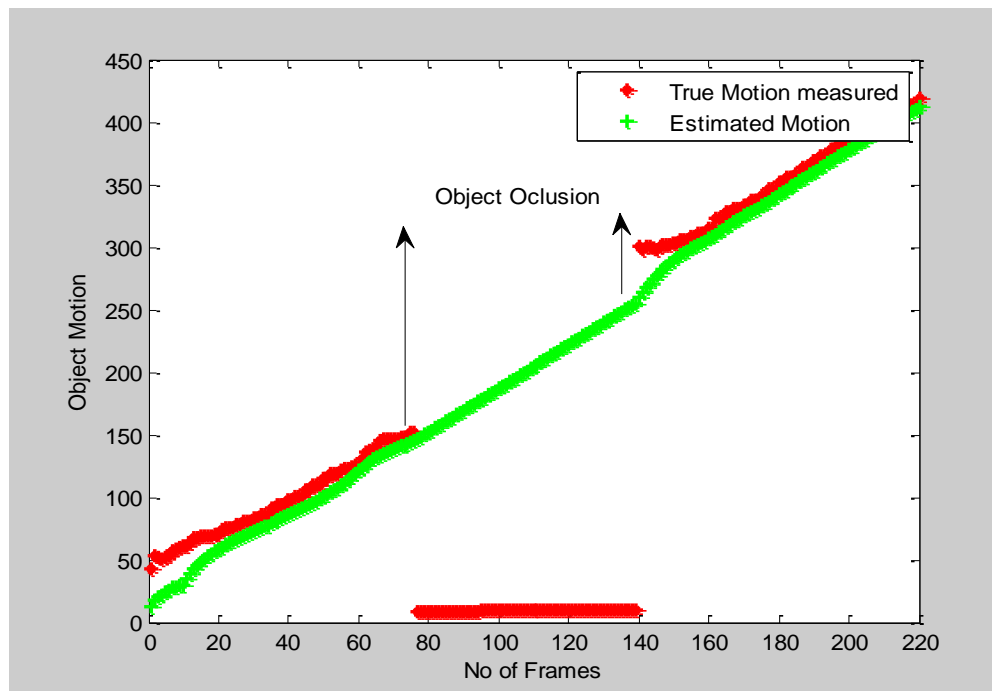
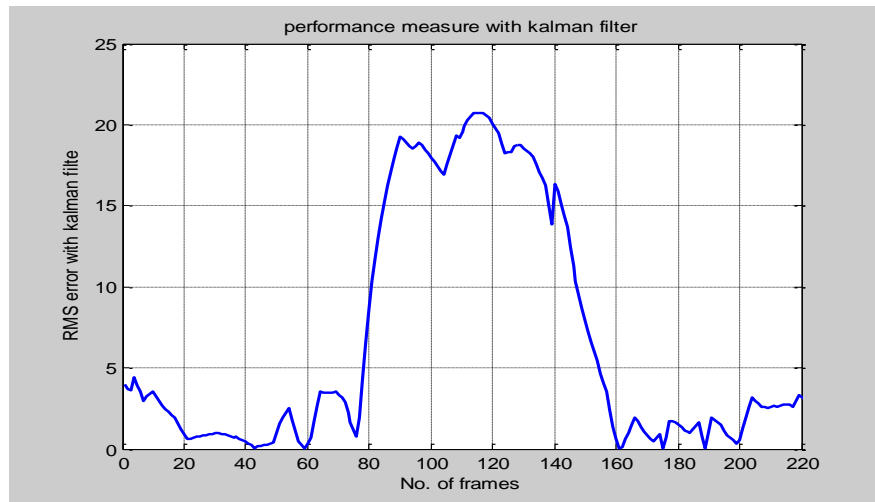
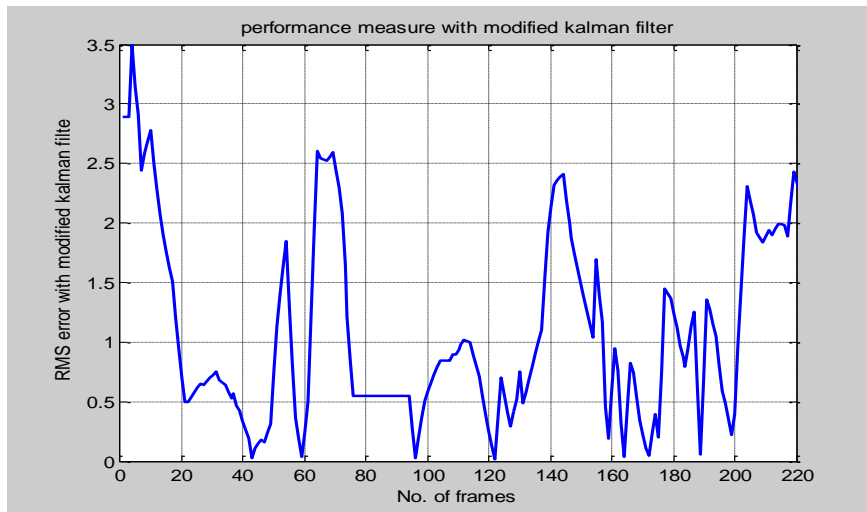


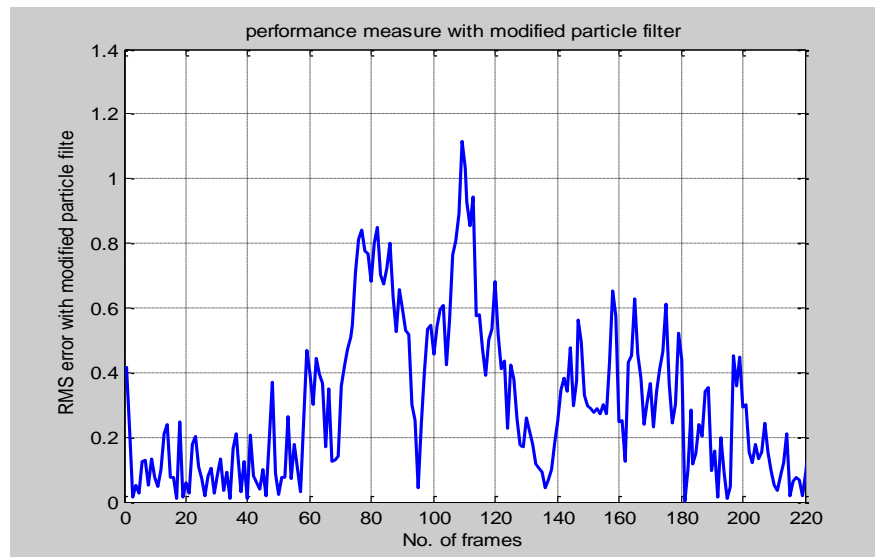
Fig. 8: Tracking Performance using the modified Particle Filter



(a)



(b)



(c)

Figure 9: a) RMS Error for the kalman filter b) RMS error for modified filter c) RMS error for modified particle filter.

Performance Measures for the proposed algorithms:

By far the most popular measure of estimation accuracy is root-mean-square (RMS) error[15]., defined by "23". Let the n-dimensional estimatee, its estimate, and estimation error which can be denoted by x , \hat{x} , and \tilde{x} respectively

$$RMSE(\hat{x}) = \left(\frac{1}{M} \sum_{i=1}^M \|\tilde{x}_i\|_2^2 \right)^{1/2} = \left(\frac{1}{M} \sum_{i=1}^M \tilde{x}_i^1 \tilde{x}_i^1 \right)^{1/2} \quad (23)$$

We can refer to it by the abbreviation RMSE.

The traditional kalman filter and particle filter is applied for person tracking with complete occlusion. The RMSE of tracking person is very high when occlusion occurs which is shown in Fig. 9(a). The modified kalman and particle filter algorithm is applied to the same scenario, tracking performance shows very low RMSE which are shown in Fig's. 9(b), (c). Also it is observed that the modified kalman and particle filter algorithms have shown fast acquisition of the image for tacking after occlusion is crossed.

VI. CONCLUSIONS AND FUTURE SCOPE

In this paper, we have proposed two modified algorithms for visual object tracking based on kalman and Particle Filter algorithms. When the target is occluded largely or totally, the proposed methods are able to track the object much faster than the traditional methods. From the experimental results, it is observed that the system can be able to recover much faster from occlusion tracking loss and also the proposed algorithms have increased the robustness of the tracking system in visual object tracking. We tested the system with a variety of video data and the performance measure RMSE shown very satisfactory results. We can extend our work in multiple possible ways such as for learning based tracking, multiple objects tracking etc.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their careful reading of this paper and for their helpful comments.

REFERENCES

- [1] Kamarul Hawari Bin Ghazali, Jie Ma, Rui Xiao, "Driver's Face Tracking Based on Improved CAMShift", IJIGSP, vol.5, no.1, pp.1-7, 2013. DOI: 10.5815/ijigsp.2013.01.01.
- [2] "Occlusion processing using particle filter and background subtraction algorithms", Guo, Tongqing; Rong, Jian; Lu, Kui; Zhong, Xiaochun Proceedings of the SPIE, Volume 8762, id. 87621U5.
- [3] "Particle filter-based visual tracking with a first order dynamic model and uncertainty adaptation", Alberto Del Bimbo, Fabrizio Dini, Computer Vision and Image Understanding 115 (2011) 771–786.

- [4] "Software environments for implementing the particle filter with dynamic models", G. Quesnel, H. Raynal, E. Ramat, and D. Makowski, , pp. 2001–2002, 2006.
- [5] "An introduction to the Kalman Filter", Greg Welch, and Gary Bishop. , Technical Report TR 95-041, University of North Carolina, Department of Computer Science, 1995.
- [6] "Office Robot Localization with a Nearest Neighbor Observation Model: A Comparison of Different Particle Filter Techniques", Erwin J.T. Buijs, Informatics Intelligent Autonomous Systems Faculty of Science (FNWI), University of Amsterdam (UvA), Netherlands. August 8, 2003.
- [7] <http://www.cvip.louisville.edu/wwwcvip/research/researchMe/tools.htm>.
- [8] F. Gustafsson, "Particle Filter Theory and Practice with Positioning Applications," IEEE Aerospace and Electronic Systems Magazine, vol. 25, no. 7, pp.53-81, July, 2010.
- [9] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," IEEE Trans.on Signal Processing, vol. 50, no. 2, February, 2002.
- [10] "Tutorial: Rao-Blackwell Particle Filtering", www.ai.mit.edu/courses/6.834J-f01/john_tutorial.doc.
- [11] http://www.juergenwiki.de/work/wiki/doku.php?id=public:particle_filter.
- [12] B. Ristic, S. Arulampalam, N. Gordon, "Beyond the Kalman Filter: Particle Filters for Tracking Applications", Artech House Publishers, 2004.
- [13] D. Simon, Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches. pp. 1–530.
- [14] "Position Estimation And Position and Map Estimation Via Particle Filtering", Stephen Bernsee, thesis report, Seth Hutchinson..
- [15] "Measures of Performance for Evaluation of Estimators and Filters", X. Rong Li Zhanlue Zhao, Proceedings of SPIE Conference on Signal and Data Processing of Small Targets, San Diego, CA, USA, July-August 2001. (4473-61).

Authors' Profiles



G.Mallikarjuna Rao is currently working as scientist in Research Centre Imarat (RCI), Defense Research Development Organisation, India. He has 14 years of experience. His research interests include Target Tracking, Image processing, Pattern recognition and Signal Processing.

He guided several M.Tech and B.Tech Projects. He published more than 10 research papers in international journals & conferences



Siva Prasad Nandyala received the B.Tech Degree in Electronics & Communication Engineering from JNTU Hyderabad and Master of Technology in Systems and Signal Processing Electronics from JNTUCEH. Presently he is a research scholar in the department of ECE, NIT Warangal.

He was worked in WIPRO technologies in VLSI domain. He is having more than 10 publications. His areas of interest include speech processing, VLSI and RF design.



Dr. Ch. Satyanarayana is Professor in Computer science and Engineering Department at Jawaharlal Nehru Technological University Kakinada, Kakinada. He has 13 years of experience. His area of interest is on Image processing, Database Management Systems, Speech Recognition, Pattern recognition and

network security.

He published more than 30 research papers in international journals. He published more than 30 research papers in international conferences.

How to cite this paper: G.Mallikarjuna Rao, Siva Prasad Nandyala, Ch.Satyanarayana, "Fast Visual Object Tracking Using Modified kalman and Particle Filtering Algorithms in the Presence of Occlusions", *IJIGSP*, vol.6, no.10, pp.43-54, 2014.DOI: 10.5815/ijigsp.2014.10.06