Particle Filter Based On Joint Color Texture Histogram For Object Tracking

Amr M. Nagy *, Ali Ahmed †, Hala H. Zayed *
* Faculty of Computers and Informatics
Benha University, Banha ElGadeda, Benha, Egypt,
amr.nagy @ fci.bu.edu.eg
hala.zayed @ fci.bu.edu.eg
† Faculty of Computers and Information,
Menofia University, Shubin el Kom, Menufia, Egypt
ali.ahmed@ci.menofia.edu.eg

***Abstract*— Particle filter has grown to be a standard tool for solving visual tracking problems in real world applications. One of the critical tasks in object tracking is the tracking of fast-moving objects in complex environments, which contain cluttered background and scale change. In this paper, a new tracking algorithm is presented by using the joint color texture histogram to represent a target and then applying it to particle filter algorithm called PFJCTH. The texture features of the object are extracted by using the local binary pattern (LBP) technique to represent the object. The proposed algorithm extracts effectively the edge and corner features in the target region, which characterize better and represent more robustly the target. The experiments showed that this new proposed algorithm produces excellent tracking results and outperforms other tracking algorithms.**

Keywords*— Object tracking; Particle filter; Local binary pattern; Color histogram.

I. INTRODUCTION

Visual tracking is one of the hot topics in the field of computer vision. It is the technology used to generate the trajectories of the moving objects by computing its motion in a sequence of images. Tracking moving objects in video sequences have very wide range of applications in video surveillance [1], military, industrial, security, intelligent transportation, human-computer interfaces, medical and other fields. Tracking algorithms can be roughly classified into two categories: deterministic methods and stochastic methods. Deterministic methods typically track the object by performing an iterative search for a similarity between the template image and the current one. On the other hand, the stochastic methods use the state space to model the underlying dynamics of the tracking system.

There were many approaches which have been proposed to improve the performances of target tracking and achieved significant improvement in the past decades. Such as background subtraction [2] inter-frame difference [3], optical flow [4], skin color extraction [5] which utilize the deterministic methods and Kalman filter [6], particle filter [7], [8], [9], [10], [11], hybrid blob and particle filter tracking approach for robust object tracking, object tracking using hybrid mean shift, particle filter and hybrid Iterated Kalman particle filter and a robust approach for object tracking based on particle filter and optimized likelihood [12], [13], [14],[15] respectively, which utilize stochastic methods.

Probabilistic methods have become popular among many researchers. The Kalman filter is a common approach for dealing with target tracking in a probabilistic framework, but it cannot resolve a tracking problem where the model is nonlinear and non-Gaussian. The extended Kalman filter can deal with this problem, but still has a problem when the nonlinearity and non-Gaussian cannot be approximated accurately.

Recently, the particle filter method, a numerical method that allows finding an approximate solution to the sequential estimation has proven very successful for nonlinear and non-Gaussian estimation problems. It approximates a posterior probability density of the state such as the object position by using samples which are called particles. An important issue in particle filtering is the selection of the proposal distribution function. In general, it is hard to design such proposals. Now many proposed distributions have been proposed in the literature. For example, the prior, the EKF Gaussian approximation and the UKF proposal are used as the proposal distribution for particle filter [13], [14].

Texture analysis has gained lots of popularity in recent years due to its wide range of uses in industry and different machine vision applications, for instance classification of different materials or visual inspection of material surfaces. Thus different operators have been introduced over the decades to enhance texture analysis, e.g., co- occurrence matrices [16] and polarograms [17]. One of the most successful operators in this field is the local binary patterns (LBP) [18]. It is grey-scale invariant and fast to compute. This makes LBP a powerful means of texture analysis. A local binary pattern is widely used in areas of visual inspection, image and video retrieval, aerial image analysis, environment modeling, biomedical image analysis, and biometrics. It is successfully applied for example to face and gender classification [19], as well as background subtraction in tracking problems [20]. Different extensions exist to LBP. Ojala et al. [21] introduced a multi- resolution extension and later they enhanced it furthermore by finer quantization of angular space using uniform patterns. LBP can be modified easily to adapt to the

978-1-4799-7069-8/14/$31.00 ©2014 IEEE

problem of interest. These modifications can be a simple threshold modification [22] or a more sophisticated approach of incorporating feature distributions.

Thus, in this study, the proposed algorithm is presented by using a particle filter with joint texture histogram features, which will lead to increase the localization accuracy of the objects in complex environment.

II. VISUAL TRACKING SYSYTEM

The visual tracking system is based on the Condensation algorithm [3], a sequential Monte Carlo method also known as particle filter or Sampling Importance Resampling (SIR) filter. The PF algorithm belongs to the filtering and data association class of tracking algorithms. PF solves the tracking problem based on the state equation

$$x_t = f_t(x_{t-1}, v_t), \tag{1}$$

Which corresponding to transition model and the measurement equation is

$$z_t = h_t(x_t, n_t), \tag{2}$$

where f_t and h_t are non-linear and time-varying functions. $\{v_t\}, t = 1,2,\ldots$ and $\{n_t\}, t = 1,2,\ldots$ are assumed to be independent and identically distributed stochastic processes. The problem consists in calculating the pdf $p(x_t|z_{1:t})$ at each time instant t.

A distribution p(x) of the state of the tracked object is approximated by maintaining a set of weighted particles (samples) $S_t = \{s_t^j\}, t \in \{1,\ldots,J\}$ over time, where each particle $s_t^j = (x_t^j, w_t^j)$ consists of its state vector x_t^j and an importance weight w_t^j. The set of particles is updated from one frame to the next by the following recursive procedure:

First, a new sample set S_t is drawn with replacement from the previous set S_{t-1}, where a sample s_{t-1}^i from the old set is chosen with probability proportional to its weight w_{t-1}^i.

Second, for each sample a new state x_t^j is determined by sampling from the motion model $p(x_t|x_{t-1} = x_{t-1}^i)$, and finally the measurement of the new frame z_t^i integrated by updating the importance weights w_t^j with the likelihood of the observation, i.e., $w_t^j = p\left(z_t|x_t = x_t^j, z_0, z_1, \ldots, z_{t-1}\right)$.The likelihood depends on all frames $z_0, \ldots, z_{t-1}$because the observation model is adapted over time. In case of a static model we have $w_t^j = p\left(z_t|x_t = x_t^j, z_0\right)$.

The observation model is the core of our approach. Before we present it in detail in Sec. III, we will first briefly describe the proposed algorithm particle filter based on JOINT color texture histogram algorithm (PFJCTH).

A. *Color histogram for target representation*

Color histograms are one of the widely used forms to represent target model in object tracking. Usually a target is defined as a rectangle or an ellipsoidal region in the frame and we compute the features inside them. Color histograms are applied as they are robust to partial occlusion, rotation, scale invariant and computationally efficient. They are also flexible in the types of object that they can be used to track, including non-rigid object (e.g. people) and rigid object (e.g. cars). Denote by $\{x_i^*\}$ the normalized pixels in the target region, which has n pixels. The probability of a feature u, which is actually one of the m color histogram bins, in the target model is computed.

$$\hat{q} = \{\hat{q}_u\}_{u=1,\ldots,m},$$

$$\hat{q} = C\sum_{i=1}^{n} K(\|x_i^*\|^2)\delta[b(x_i^*) - u], \tag{3}$$

where $\hat{q}$ is the target model, $\hat{q}_u$ is the probability of the uth element of $\hat{q}$, δ is the Kronecker delta function, $b(x_i^*)$ associates the pixel x_i^* to the histogram bin, k(x) is an isotropic kernel profile, and constant C is

$$C = 1\Big/\textstyle\sum_{i=1}^{n} K(\|x_i^*\|^2), \tag{4}$$

Similarly, the probability of the feature $u = 1, 2, \ldots, m$ in the target candidate model from the target candidate region centered at position y is given by

$$\hat{P}(y) = \left\{\hat{P}_u(y)\right\}_{u=1,\ldots,m},$$

$$\hat{P}_u(y) = C_h \sum_{i=1}^{n_h} K\left(\left\|\frac{y - x_i}{h}\right\|^2\right)\delta[b(x_i) - u], \tag{5}$$

where $\hat{P}(y)$ is the target candidate model, $\hat{P}_u(y)$ is the probability of the uth element of $\hat{P}(y)$ $\{x_i\}_{i=1,..,n_n}$ are pixels in the target candidate region centered at y, h is the bandwidth and C_h is the normalized constant

$$C_h = 1\Big/\textstyle\sum_{i=1}^{n_h} K\left(\left\|\frac{y - x_i}{h}\right\|^2\right) \tag{6}$$

In order to calculate the likelihood of a candidate we need a similarity function which defines a distance between the model and the candidate. A metric can be based on the Bhattacharyya coefficient, defined between two normalized histograms p(y) and q as

$$\rho[p(y), q] = \sum_{u=1}^{m} \sqrt{p_u(y) q_u}. \tag{7}$$

Hence we define the distance as

$$d[p(y), q] = \sqrt{1 - \rho[p(y), q]} \tag{8}$$

B. *Color and Texture for target representation*

The local binary pattern (LBP) operator is an image operator which transforms an image into an array or image of integer labels describing small-scale appearance of the image. These labels or their statistics, most commonly the histogram, are then used for further image analysis.

The original version of the local binary pattern operator works in a 3 × 3 pixel block of an image. The pixels in this block are thresholded by its center pixel value, multiplied by powers of two and then summed to obtain a label for the center pixel. As the neighborhood consists of 8 pixels, a total of

$2 \times 8 = 256$ different labels can be obtained depending on the relative gray values of the center and the pixels in the neighborhood.

The LBP operator is defined as follows:

$$LBP_{P,R}(x_i, y_i) = \sum_{p=0}^{p-1} \varphi(g_p - g_i)^{2^p}, \quad (9)$$

where P indicates target area of pixels, R represents the target pixel radius, g_i is the grey at the center pixel (x_i, y_i) and g_p is the grey value of the adjacent pixels around center point g_i in the region. The function $\varphi(x)$ can be defined as:

$$\varphi(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \quad (10)$$

Figure 1 shows an example of LBP texture model.

Example

6	5	2
7	6	1
9	8	7

Threshold

1	0	0
1		0
1	1	1

Weights

1	2	4
128		8
64	32	16

Fig. 1: An example of computing $LBP_{8,1}$ texture model in 3×3 neighrbood.

In example LBP operator labels the pixel in an image by thresholding its neighborhood with the center value and considering the result as a binary number (binary pattern). So that the Pattern = 11110001 and LBP = 1+16+32+64+128 = 241.

However, the LBP may construct a sparse histogram, so the histograms lose statistical significance, because the LBP operator produces a more binary mode, but actually the number of pixels in the target area is relatively small. So a LBP uniform mode can be proposed, which is defined as:

$$LBP_{P,R}^{riu2} = \begin{cases} \sum_{p=0}^{P-1} \varphi(g_p - g_i)\, 2^p & , U(LBP_{P,R}) \leq 2 \\ P+1 & otherwise \end{cases} \quad (11)$$

where function $U(LBP_{P,R})$ calculates the hop variables in the sequence of the binary model, where the binary can be changed from "0" to "1" or from "1" to "0". A local binary pattern is called uniform if its uniformity measure is at most 2. For example, the patterns 00000000 (0 transitions), 01110000 (2 transitions) and 11001111 (2 transitions) are uniform whereas the patterns 11001001 (4 transitions) and 01010011 (6 transitions) are not. The function $U(LBP_{P,R})$ can be described as:

$$U(LBP_{P,R}) = |\varphi(g_{p-1} - g_i) - \varphi(g_0 - g_i)| + \sum_{P=1}^{P-1} |\varphi(g_p - g_i)\varphi(g_{p-1} - g_i)| \quad (12)$$

In uniform LBP mapping there is a separate output label for each uniform pattern and all the non-uniform patterns are assigned to a single label (P+1).

One of the drawbacks of the LBP operator is that the thresholding operation in comparing the neighboring pixels could make it sensitive to noise. Heikki¨a and Pietik¨ainen [19] are able to overcome the disadvantage of the LBP operator by modifying the thresholding strategy which replace the term $\varphi(gp - gi)$ in Eqs. (9), (11) and (12) with $\varphi(gp - gi + a)$. It's obvious that the greater the value |a| is, the higher fluctuations in pixel values are allowed without affecting much the thresholding result.

In $LBP_{P,R}^{riu2}$ the subscript represented using the operator in a (P, R) neighborhood is riu2 stands for using only uniform patterns. If P=8 and R =1then there are nine uniform patterns in $LBP_{8,1}^{riu2}$. V. Hlavac and R. Boyle M. [23] consider each of the $LBP_{8,1}^{riu2}$ uniform patterns as a micro-texton. There are two categories of uniform patterns one for major uniform pattern which correspond to edges, line ends and corners {2,3,4,5,6} and one for minor uniform patterns which correspond to spots and flat areas{0,1,7,8}.

To represent the target model, we use the thresholding strategy proposed by [19] and extract the main uniform patterns of the target by the following equation.

$$LBP_{8,1}^{riu2} = \begin{cases} \sum_{p=0}^{7} \varphi(g_p - g_i + a) & U(LBP_{8,1}) \leq 2 \text{ and} \\ & \sum_{p=0}^{7} \varphi(g_p - g_i + a) \in \{2,3,4,5,6\} \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

Equation (13) extracts the main LBP features of a target. It's obvious that the major features are more important minor features to represent the target. So that only the pixels corresponding to the main LBP features are extracted and then use the color and texture model features of these pixels to model the target appearance model.

Lei Zhang, David Zhang [22] proved that target representation with color histogram and LBP histogram could effectively removes the smoothing background and extracts the main features. It reduces greatly the interferences induced by minor uniform patterns, which come from mainly the smooth background and noise in the target area.

III. The Proposed Tracking Algorithm with the Joint Color-Texture Histogram (PFJCTH)

LBP patterns are extracted by (13) to represent the target in RGB color space and embed it into the particle filter tracking algorithm. We use (3) to calculate the color and texture distribution of the target model $\hat{q}$ of the target region, in which $u = 8 \times 8 \times 8 \times 5$. The first three dimensions $(i.e. 8 \times 8 \times 8)$ represent the quantized bins of color channels and the fourth dimension (i.e. 5) is the bin of the modified LBP texture patterns by (13). Similarly, the target candidate model $\hat{p}(y)$ is calculated with (5). The whole tracking algorithm is summarized as follows.

A. *Proposed algorithm (PFJCTH)*

Input: Target state x_{t-1}(previous frame)

Output: Target state x_t (current frame)

1. Initialization
2. K=0,initialize tracked region at x_0
3. Generate N samples (Particles) $\{x_n^{(n)}\}$, n =1,2,…,N from the initial distribution $p(x_0)$
4. Initialize weights $w_0^i = 1/N$
5. Resampling weights
6. For t =1: T Frames
7. For n =1,…,N
8. Compute the Joint Color-Texture Histogram of tracked object $q\hat{}$
9. Importance sampling for each particles
10. Propagate each sample according to the system model of eq.2
11. Calculate the Joint Color-Texture Histogram for the candidate target model $P\hat{}$
12. Calculate the Bhattacharya Coefficient
13. $d = \sqrt{1 - \rho[p, q]}$
14. Calculate the weight
15. $w^{(i)} = \frac{1}{\sqrt{2\pi\sigma}} e^{\left(\frac{-d^2}{2\sigma^2}\right)}$
16. Estimate the position of x_t according to the mean estimate
17. $\bar{x} = \sum_i^N x_t^i \ w_t^i$

IV. EXPERIMENTAL RESULTS

The performance of proposed PFJCTH tracker is evaluated on selected videos from PETS 2001 and PETS 2000 to exemplify different conditions.

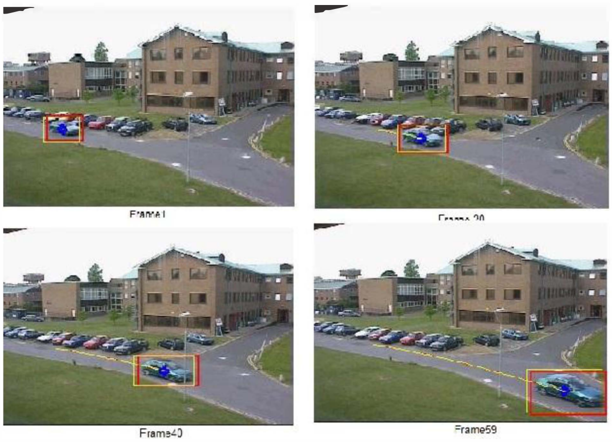

Fig. 2: The tracking result of PFJCTH tracker in the sequence 1 of car video from PETS 2001 Dataset.

The target region is obtained from color and texture of the distribution in which $u = 8 \times 8 \times 8 \times 5$ The first three dimensions (i.e. 8 ×8 ×8) represent the quantized bins of color channels and the fourth dimension modified LBP texture patterns by Experiments are conducted to test for its ability to handle scale changing and velocity changes as well as overlapping with like colored background.

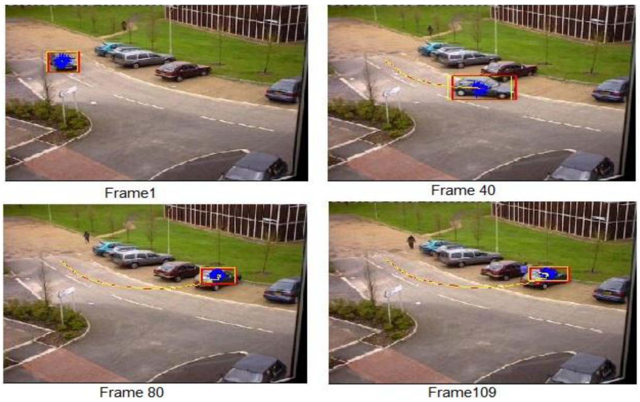

Fig. 3: The tracking result of PFJCTH tracker in the sequence 2 of car video from PETS 2000 Dataset

These experiments aims to test and evaluate the performance of PFJCTH tracker in tracking the target car which continuously experiences scale changings and velocity changes, as shown in Fig.4 and Fig.5. These results demonstrated successive tracking of car by PFJCTH tracker. The success of this tracker in dealing with scale changings and velocity changes of fast moving object supports the utilization of the tracker in tracking random motion object.

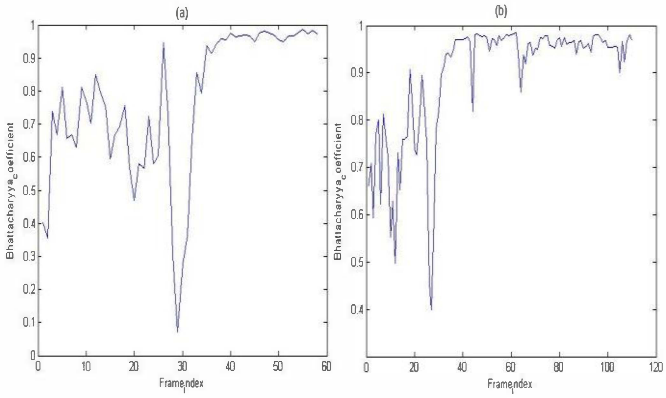

Fig. 4: Bhattacharyya coefficient at each frame for the PFJCTH tracker for video sequence1 (a) and video sequence 2 (b).

The Bhattacharyya distance between the tracked candidate object region and the reference object region is defined by (8). Fig.6. Higher Bhattacharyya coefficient indicates that the estimated target candidate distribution is matched with the target model distribution with higher similarity. It can be observed that the PFJCTH tracker can achieve sufficiently high coefficient in the most of the frames.

From Fig.7 and Fig.8 the tracking trajectory of the proposed system is much closer to the true trajectory than that obtained by MS-Joint Color texture histogram as shown in two sequences. The tracking trajectory obtained by MS-Joint color texture histogram has a large diffusion, due to scale change and background clutter.

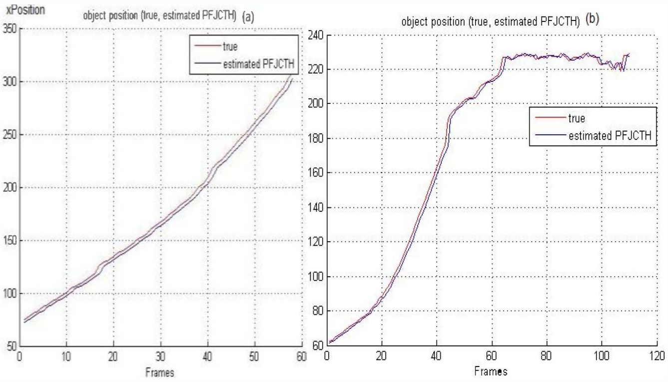

Fig. 5: Tracking trajectory for the video sequence 1 (a) and video sequence 2 (b) using PFJCTH tracker.

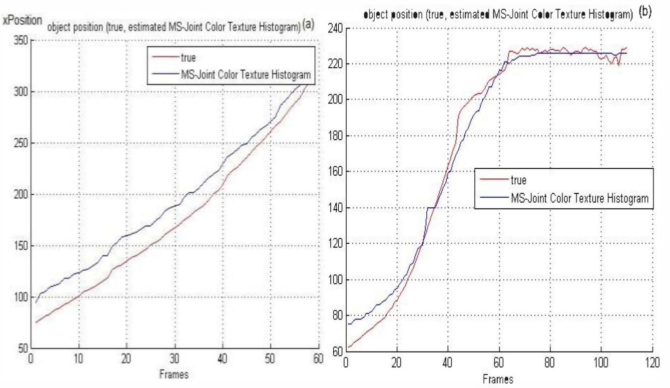

Fig. 6: Tracking trajectory for the video sequence 1 (a) and video sequence 2 (b) using MS-Jiont- color texture histogram tracker tracker.

V. PERFORMANCE EVALUTION

To further evaluate the performance of the proposed algorithm and to compare with the existing trackers, the root mean square (RMSE) error was utilized for quantitative evaluation.

TABLE I. ESTIMATION OF MEANS OF RMSE.

Sequence	RMSE	
	PFJCTH	***MS - Color-joint texture histogram***
Seq.1	4.3	12.9
Seq.2	2.8	9.6

The performance of this proposed system was measured using the mean of the RMSE, in case of PFJCTH and MS with color-joint texture histogram. Table.1 shows that the RMSE is 4.3 in case of PFJCTH and 12.9 in case of MS with color joint texture histogram for video sequence1; also the RMSE is 2.8 and 9.6 for video sequence 2 respectively. These results indicate that the proposed algorithm shows a better performance and tracking accuracy.

CONCLUSION

The developed PFJCTH tracker produces satisfactory tracking performance and produces a small RMSE. To improve the robustness of target representation, the PFJCTH object tracker is demonstrated with using color- joint texture histogram information as the feature representations of the target. A mask of the target is formed based on its five major uniform $LBP_{8,1}^{riu2}$ texture patterns and then the target is represented by using its color and texture features within the mask. The proposed target representation model effectively extracts the edges and corners, which are important and robust features, of the object while suppressing the smooth background features. As a result, this tracker is capable of tracking scale change of the object in complex environment.

REFERENCES

[1] Larry S. Davis, Kyungnam Kim, "Object Detection and Tracking for Intelligent Video Surveillance," in Multimedia Analysis, Processing and Communications, 2011, pp. 265-288.

[2] Y. LIU and A. & Xu Guangyou, Haizho, "Moving object detection and tracking based on background subtraction," Proceeding of Society of Photo-Optical Instrument Engineers, vol. 4554, pp. 62-66, 2001.

[3] R. Collins et al., "System for video surveillance and monitoring," Technical report CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University, 2000.

[4] D. Meyer and J. & Niemann, H Denzler, "Model based extraction of articulated objects in image se-quences for gait analysis," in Proceeding of IEEE Int. Conf. Image Proccessing, 1998, pp. 78-81.

[5] S. Phung and D. & Bouzerdoum, A., Chai, "Adaptive skin segmentation in color images," in Proceeding of IEEE International Conference on Acoustics, Speech and Signal Processing, 2003, pp. 353-356.

[6] T. & Chellappa, R., Broida, "Estimation of object motion parameters from noisy images," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, pp. 90-99, 1986.

[7] Isard M. & Blake A., "CONDENSATION-Conditional Density Propagation for Visual Tracking," Intl. Journal of Computer Vision, vol. 29, pp. 5-28, 1998.

[8] B. Ristic and S. & Gordon, N., Arulampalam, "Beyond the Kalman filter: Particle filters for tracking applications," Artech House, 2004.

[9] Gao Ming. Xu Fen, "Pedestrian Tracking Using Particle Filter Algorithm," in International Conference on Electrical and Control Engineering, 2010.

[10] Zhaoyi Peng, Xiaojun Deng, Shifeng Li. Zhiqiang Wen., "Particle Filter Object Tracking Based on Multiple Cues Fusion," Advanced in Control Engineering and Information Science, pp. 1461-1465, 2011.

[11] ZHANG Qingqinga, GAO Tengfeia, QU Zhonga, "Moving Object Tracking Based on Codebook and Particle Filter," in International Workshop on Information and Electronics Engineering, 2012, pp. 174-178.

[12] Zulaikha Kadima, Kim Meng Lianga, Mei Kuan Lima Sze Ling Tanga, "Hybrid Blob and Particle Filter Tracking Approach for Robust Object Tracking," in IProceedings of the International Conference on Computational Science, 2010, pp. 2549-2557.

[13] Naeem. Asad and Pridmore. Tony, "Object Tracking Using Hybrid Mean Shift and Particle Filter Algorithms: An indepth discussion on computer vision object tracking algorithms," LAP LAMBERT Academic, 2012.

[14] Amr M. Nagy, Ali Ahmed, Hala H. Zayed "Hybrid Iterated Kalman Particle Filter for Object Tracking Problems," in IProceedings of the International Conference on Computer Vision Theory and Applications, 2013, pp. 375-381.

[15] Amr M. Nagy, Ali Ahmed, Hala H. Zayed "A robust approach for object tracking based on particle filter and optimized likelihood ," International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS), vol. 1, no. 7, pp. 54-61, 2014.

[16] S. A. Johns, and J. K. Aggarwal. L. S. Davis, "Texture analysis using generalized cooccurence matrices," IEEE Trans. Pattern Anal. Mach. Intell, vol. 1, pp. 251–259, 1979.

[17] L. S. Davis. Polarograms, "A new tool for image texture," Pattern Recognition, vol. 13, pp. 219–223, 1981.

[18] J. Viertola, and M. Pietikäinen. T. Mäenpää, "Optimising colour and texture features for realtime," Pattern Analysis and, vol. 6(3), pp. 169-175, 2003.

[19] A. Hadid and M. Pietikäinen., "Combining appearance and motion for face and gender recognition from videos," Pattern Recognition, vol. 42(11), pp. 2818-2827, 2009.

[20] Marko Heikkilä and Matti Pietikäinen, "A texturebased method for modeling background and detecting moving objects," IEEE Trans. Pattern Anal.Mach. Intell., vol. 28(4), pp. 431-438, 2006.

[21] Matti Pietikäinen, and Topi Mäenpää Timo Ojala, "Multiresolution gray scale and rotation invariant texture classification with local binary patterns," IEEE Trans. Pattern Anal. Mach. Intell., vol. 24(7), pp. 971–987, 2002.

[22] Lei Zhang, David Zhang, and Chengke Wu Jifeng Ning, "Robust object tracking using joint color texture histogram," International Journal of Pattern Recognition and Artificial Intelligence, vol. 23(7), pp. 1245–1263, 2009.

[23] V. Hlavac and R. Boyle M. Sonka, Image Processing, Analysis and Computer Vision, 3rd ed.: Thomson, 2007.

[24] V. Ramesh and P. Meer, D. Comaniciu, "Kernel-based object tracking," IEEE Trans.Patt. Anal. Mach. Intell, vol. 25, no. 5, pp. 564-575, 2003.

[25] P. Scheunders and D. Dyck, G. Wouwer, "Statistical texture characterization from," IEEE Trans. Imag. Process., vol. 8, no. 4, pp. 592-598, 1999