Target Tracking Using Color Based Particle Filter

Amir Mukhtar, Likun Xia
Centre for Intelligent Signal and Imaging Research (CISIR), Department of Electrical and Electronic Engineering,
Universiti Teknologi PETRONAS, Seri Iskandar 31750 Tronoh, Perak, Malaysia
amir22j@gmail.com, likun_xia@petronas.com.my

***Abstract* — A robust and efficient visual target tracking algorithm using particle filtering is proposed. Particle filtering has been proven very successful in estimating non-Gaussian and non-linear problems. In this paper, particle filter with color feature estimated the target state with time. Color feature being scale and rotational invariant, have showed robustness to partial occlusion and computationally efficient. The performance is made more robust by choosing the different (YIQ) color scheme. Tracking has been performed by comparison of chrominance histograms of target and candidate positions (particles). The Color based particle filter tracking often leads to inaccurate results when light intensity changes during a video stream. Furthermore, background subtraction has been used for size estimation of target. The qualitative evaluation of proposed algorithm is performed on several real world videos. The experimental results demonstrated that the proposed algorithm can track the moving objects well under illumination changes, occlusion and moving background.**

***Index Terms*—Tracking, particle filter, histogram, corner points, occlusion, illumination.**

I. INTRODUCTION

Target tracking [1] is an important element of computer vision applications such as surveillance [2], human computer interaction [3], obstacle avoidance system [4] and video compression and communication [5]. The main function of tracking is to determine the position, number and other motion parameters of the object in motion. Filter is the most fundamental block of any tracking system and its role is to recursively estimate the objects position with time. In vision based tracking, a target is initially specified and then, it is searched in coming frames. This search can be performed in variety of ways depending upon strengths and weaknesses of algorithms used. Tracking usually has two components: target representation and localization/filtering. Following are some of the techniques which are used for video tracking,

- Background subtraction [6]
- Mean-shift tracking [7]
- Kalman filter [1, 8]
- Particle filtering [9, 10]

One of the simplest of methods used for tracking is the Background subtraction. It involves subtraction of current frame from reference frame and yields the moving objects in a video stream. Limitation of this technique is that camera must be stationary else it becomes difficult to isolate the background. Mean-shift algorithm is another way to detect and track the movement of an object. In this iterative technique, the histograms of original object in the current frame and that of candidate regions in next frame are compared. The goal is to maximize the correlation between the two histograms. Kalman filtering is also a method used to track a target in video, it uses Bayesian framework for tracking purposes. The limitation is that the motion model of system must be linear and noise in the system is Gaussian. But if the system model is non-linear and the process noise is non-Gaussian then one has to opt Particle filter (PF).

PF is a numerical method that determines an approximate solution to the sequential estimation and has been successfully applied in single and multiple objects tracking problems. It is an effective technique of solving nonlinear, non-Gaussian, problems by implementing Bayesian approximation using Monte Carlo simulations [11]. In PFs, the required posterior density function is represented by a set of random samples (particles) with associated weights. Finally, the state is estimated on the basis of magnitudes of weights associated with each sample. Particle filters use several features to compute weights and motion estimation. These features have their own advantages and disadvantages e.g., color feature shows robustness against partial occlusion and noise but has poor performance with background of matching colors and illumination changes. On the other hand, edge feature is robust against illumination changes but fails in cluttered environment [12].

Since particle filtering is a numerical approach using Monte Carlo methodology, it requires state of the art computational machines. To process a video, it is desired to propose an algorithm which is executed in minimum time. For robust tracking, a particle filter using both color and edge or contour features is used [12], which almost doubles the processing data and increases the processing time. Instead of using two features, we stick with one feature with slight modification in its utilization. Color information is used in NTSC format [13], which eliminates the effect of illumination changes. It is also time efficient due to the fact that we are only required to perform the calculation and comparison of histograms of two components instead of three (in case of RGB). Furthermore, after getting the estimated position from particle filter method with the aid of pre-defined value of size of the target, a search window is constructed in the proximity of target. New size of the target is determined by locating the extreme corners of the object in search window. This unique combination of color particle filtering and background subtraction results in efficient, precise and robust target tracking.

978-1-4799-4653-2/14/$31.00 © 2014 IEEE

The paper is organized as follows: Section II presents theoretical account of particle filter and tracking procedure based on the color cue. In Section III, a tracking algorithm is proposed based on particle filter using color feature. Section IV contains experimental evaluation of proposed technique on real world videos and their results. Finally, conclusions are drawn in Section V.

II. PARTICLE FILTERING

PFs perform Sequential Monte Carlo Estimation (SME) based on point mass (or "particle") representation of probability densities. In statistics, basic sequential Monte Carlo (SMC) ideas in sequential importance sampling form were first introduced in 1950s [14]. These ideas were continuously explored during the 1960s and 1970s [15, 16], but their usage was restricted due to limited computational power available at that time. Further, these earlier implementations were based on normal sequential importance sampling, which causes degeneration with time. Practical implementation of particle filters increased dramatically after the significant contribution of resampling step [17] in the development of the SMC method and advancement in computational technology. Since then, a substantial increase in research areas of particle filtering [18, 19], has resulted in numerous implementations of PFs and their various applications.

Particle filter estimates the state of a system x_k (seen in (1)) that is governed by stochastic difference equations with the measurement z_k (seen in (2)).

$$x_k = Ax_{k-1} + Bu_k + w_{k-1} \quad (1)$$

$$z_k = Hx_k + v_k \quad (2)$$

The random variables w_k and v_k represent the process and measurement noise (respectively). They are assumed to be independent (of each other), white, and with normal probability distributions. The matrix A relates the previous state to the current state and matrix B relates the optional control input u to the state x whereas, H in the measurement equation relates the state to the measurement.

The general programming algorithm for particle filtering implementation is shown in Fig. 1. Initially, particles are drawn using the previous information of the system state. After making the observation(s), weights are assigned to each individual particle based and cumulative sum is calculated normalize weights. Since, particles represent the next probable state of the system so higher the normalized weight of a particle, closer it will be to the actual state of system. Next stage is to perform resampling; it eliminates samples (particles) with smaller probability and duplicates the other with higher probabilities. Mean or expected value of particles is calculated to determine the current state of the system.

A. Color Based Particle Filtering

A PF using color feature of images can be used to track moving objects in videos [20]. Visual tracking of non-rigid objects employing the color histograms have many advantages as they show robustness towards partial occlusion, are scale

$\left[\{x_k^i\}_{i=1}^N\right] = PF\left[\{x_{k-1}^i\}_{i=1}^N, z_k\right]$

FOR $i = 1:N$

– Draw $x_k^i \sim p\left(x_k | x_{k-1}^i\right)$

– Calculate $\tilde{w}_k^i = p\left(z_k | x_k^i\right)$

END FOR

Calculate total weight: $t = \text{SUM}\left[\{\tilde{w}_k^i\}_{i=1}^N\right]$

FOR $i = 1:N$

– Normalize $w_k^i = t^{-1}\tilde{w}_k^i$

END FOR

$\left[\{x_k^i, -, -\}_{i=1}^N\right] = \text{RESAMPLE}\left[\{x_k^i, w_k^i\}_{i=1}^N\right]$

END

Fig. 1. Particle Filter Algorithm

and rotation invariant and calculated efficiently. The similarity between histograms of target model and that of sample positions determines the probability of different target states. By doing so, image content is evaluated only at random particle positions which eliminates the segmentation of entire image.

In comparison to edge-based PFs, color-based PFs remain unaffected against out-of-plane rotations. Usually, PF utilize color features for initialization and importance sampling [10]. Generally, color-based particle filters give bad results due to illumination changes or when the background has matching colors with the target model. In that case, one has to use another feature for making the algorithm more robust against these changes and to track properly as it was done in [9, 12].

Addition of a new feature can surely make the results more accurate but it will make the job too difficult and complex in programming and computation will require additional time for processing. Particle filtering itself is complex and computationally heavy, so adding another feature will almost double the execution time. A solution to this problem is to make a color based PF robust enough to track the target efficiently even in illumination changes. For this, we look for alternate color-information usage of image that remains unaffected in illumination change.

B. NTSC Color Space

The NTSC is a color scheme with an advantage of having the luminance information separated from chrominance data. In this format, grey scale data or intensity level is luminance (Y) while hue (I) and saturation (Q) contain the color information of an image. YIQ values can be determined form the RGB components by using the following transformation [9],

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

where luminance (Y) ranges from 0 to 1, hue (I) from -0.596 to

+0.596, and saturation (Q) from -0.523 to 0.523. In order to make the illumination changes ineffective, only Hue and Saturation components are compared and luminance information is ignored. Another advantage of using NTSC scheme is that there is no preprocessing required in obtaining NTSC format as CCTV cameras and typical TV signals transmit the video information using this format. Furthermore, histogram comparison of two components (I and Q) is performed in a timeless manner than the time taken in case of 3 components (RGB).

C. Background Subtraction

Background subtraction is a technique of separating the foreground objects from unwanted background scene in an image. Foreground objects are defined as set of pixels or portion of image containing object of interest whose information provides valuable contribution to the task under consideration and reduces the data to be processed. In video tracking, foreground objects correspond to the moving objects in a video sequence and detection of such objects is crucial and fundamental task in computer vision research.

Image segmentation utilizes background subtraction for segmenting and extracting specific objects from videos for surveillance purposes. Development of an efficient background subtraction algorithm poses a few challenges which are: avoidance of detecting any stationary background object and shadows of foreground objects, and to remain unaffected during illumination variations. Such an effective method should respond rapidly to background changes and modify itself to the new changes happening in the background e.g., temporary displacement of a stationary object. Furthermore, it should have a fast and real time foreground detection rate for practical applications. Various methods [6] have been proposed to perform the background subtraction. A simplest method to extract background is to subtract the previous frame from the current frame to get the moving objects in the scene.

III. Methodology

Tracking of a moving object in a video was performed using a particle filter based on color features of the object. Color information was used in NTSC domain. The algorithm consists of following steps:

a) Target Identification: This step involved the target specification i.e. its color information, initial size and position. The color information of the target was collected in the form of histogram. The RGB histogram of the target was first converted to YIQ histogram before its comparison or further processing. Number of bins in histogram was set to 100 instead of RGB’s 256.

b) PF Initialization: After getting the target information, filter was initialized with 150 particles (N=150). The state vector consisted of the initial position of the target i.e. $\mathbf{X} = (x, y)$ with x and y denoting the position of target. For simplicity, the random walk model was taken as motion model (3), in which the state evolution equation was,

$$\mathbf{X}_t = A\mathbf{X}_{t-1} + \mathbf{V}_t \tag{3}$$

where, $A = I$ is the transition matrix and $\mathbf{V}_t$ is process noise.

c) Color Model and Measurement: Color feature of the target was described by its color histogram. Suppose, H_t is the NTSC histogram of the target having three components Y, I, and Q. We used I and Q for our tracking purpose, and ignored Y. Candidate regions were generated by N randomly distributed particle positions with predicted state as their mean. Histograms of the candidate regions H_c^i were compared with the original one H_t by Euclidean distance.

$$D_i = \left|H_t - H_c^i\right|, \quad \text{where } i = 1,2,3,\dots,N$$

After obtaining the distances of each region, weights were assigned to each particle using Gaussian formula (4).

$$w_i = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{D_i^2}{2\sigma^2}\right) \tag{4}$$

where σ is variance and its value was chosen arbitrarily. Greater the value of σ, larger will be the variation in magnitude of weights. Weights were then normalized to perform the resampling step.

d) Resampling: Samples with bigger weights were multiplied and replicated according to their magnitude and those with smaller weights were removed, by keeping the total number of samples unchanged.

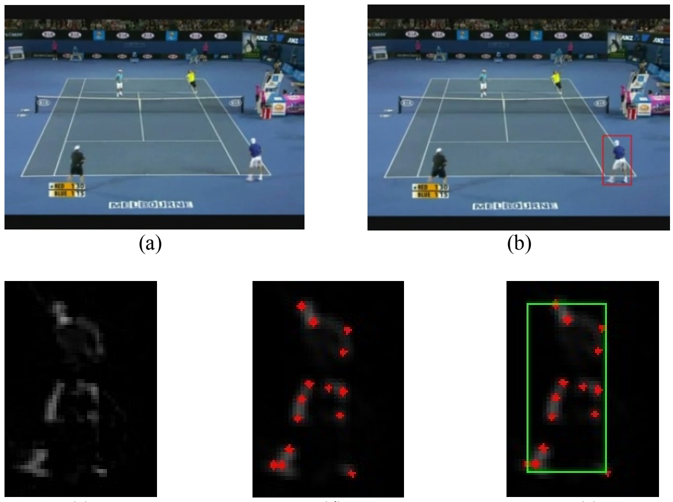
e) Position Estimation: The state (position) of the target was estimated by multiplying the particle’s position with their respective weights (5).

$$E = \sum_{i=1}^{N} w_i x_i \tag{5}$$

f) Size Estimation by Background Subtraction: At the end of particle filtering process, the point (central location) was obtained. If the object changed its size in the frame, then the new size was determined by background subtraction technique. The size of target in previous frame was utilized along with the expected position calculated by particle filter, for finding the new size of the target. A marginal value was also used to construct a search box around the target. The search box is shown in Fig. 2(b). Frame subtraction was performed only on the region inside the search box to save a fraction of the execution time. Figure 2(c) shows the result of this subtraction. Corner points of moving objects inside the search box were obtained by using corner points detection methods [21] and shown in Fig. 2(d). Minimum and maximum values of corner point co-ordinates were useful in finding the new size of target. To remove the noise or isolated points, Gaussian filter [13] was applied before finding these corner points. Figure 2(e) shows the final result.

IV. Experimental Results

The proposed PF based tracker was implemented in 32-bit MATLAB (7.12.0) and tested on a core i3 @ 2.67 GHz PC with 4GB memory. Several real life video sequences of resolution 240 × 320 were used to test the effectiveness of proposed algorithm (with N=150) and average tracking time of 0.07s per frame was recorded. Since MATLAB does not support the video reading/loading in any format other than RGB, therefore RGB to YIQ conversion needs to be done before the processing. Tracking time of 0.07s per frame was recorded (excluding RGB to NTSC conversion time).



(a) (b)

(c) (d) (e)

Fig. 2. Size estimation: (a) Frame 130; (b) Frame 131 with search box around the target; (c) Frame difference; (d) Corner detection result; (e) resultant size of target

(a) Frame 583 (b) Frame 624

(c) Frame 638

Fig. 4. Successful YIQ tracking results despite of large illumination changes

A) Tracking under the Background with Illumination Changes

In first experiment, a single person is shown walking in open air environment having variable light intensities. The tracking results based on RGB color histogram are shown in Fig. 3. Target was totally lost by the tracker when it went in the brighter region as shown in frame 638.

When the same experiment was repeated using color information in YIQ form, the tracker followed the target constantly and accurately regardless of large illumination changes. Results are shown in Fig. 4.

B) Tracking during Occlusion

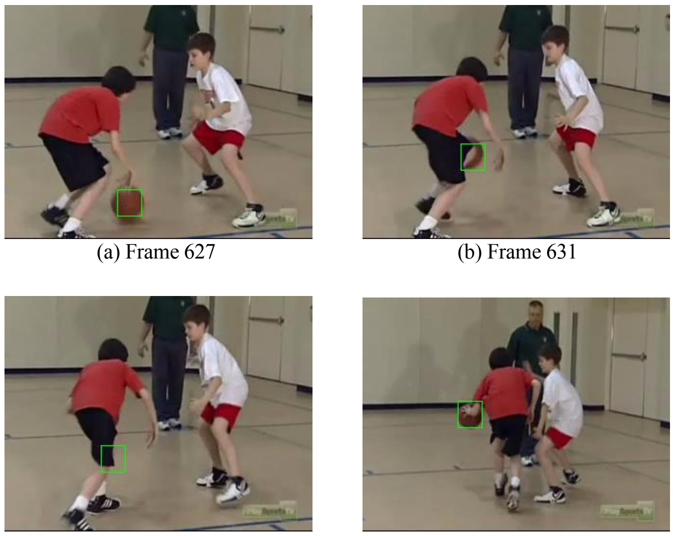
In second experiment, the proposed algorithm showed fine tracking results when occlusion occurs (see Fig. 5). The tracker temporarily lost the target (basketball) when it goes completely behind the player's body. However, once the occlusion was removed, the proposed algorithm instantly recovered the target position. Figure 5(d) is the frame in which the object re-appeared again after the occlusion. The tracker located the object in a span of 2 frames after its partial re-appearance showing its robustness.

Another case involving complete occlusion is shown in Fig. 6. Here, a man disappears from the scene by entering a room and becomes invisible for some frames. The tracker temporarily lost it but as soon as the man started coming out from the door, it immediately caught it and continued the tracking operation.

(a) Frame 583 (b) Frame 624

(c) Frame 638

Fig. 3. RGB tracking results when illumination varies



(a) Frame 627 (b) Frame 631

(c) Frame 635 (d) Frame 655

Fig. 5. Tracking results when occlusion happens

(a) Frame 53 (b) Frame 95

(c) Frame 117 (d) Frame 147

Fig. 6. Tracking results when occlusion happens

V. CONCLUSION

In this paper, a tracking approach based on the renowned Particle Filter model has been introduced to be a powerful, efficient and robust technique for various applications. The Particle Filter algorithm has been very successful in estimating a variety of probability density functions.

Its popularity can be attributed to both its versatility and easy implementation. This color based tracker effectively dealt with fast moving and non-rigid objects under different luminance conditions.

Although color histogram is a frequently used observation parameter due to its efficiency and simplicity, it is difficult to remain stable and precise when the environment is complex. We have modified the color based tracking by utilizing the NTSC color space, allowing the algorithm to become more robust and persistent. The contribution of Background Subtraction technique is also useful in locating position and size of the target. Background Subtraction itself is used for tracking purposes but here it is integrated with particle filtering to determine the target size. With the fusion of these two approaches, the proposed algorithm accomplishes much improved performance under different illumination conditions and occlusion.

The execution time in each experiment was also calculated and it is observed that this time depends on three main factors. These factors are size of target, no. of particles/samples, video resolution. Currently, we used a simple kinematic system model to generate the sample set. By replacing this straight forward model with a learned/trained motion model, tracking quality could be improved considerably. There may be some imperfections in the algorithm and our future work can significantly figure it out. The use of edge information can be useful in the scenarios of matching colors. Moreover, the proposed algorithm can easily be extended to multiple objects tracking in single and multiple camera environments.

REFERENCES

[1] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman filter: Particle filters for tracking applications*. Boston London: Artech House, 2004.

[2] J. Sherrah, B. Ristic, and N. J. Redding, "Particle filter to track multiple people for visual surveillance," *Computer Vision, IET,* vol. 5, pp. 192-200, 2011.

[3] C. Colombo, A. Del Bimbo, and S. De Magistris, "Human-computer interaction based on eye movement tracking," in *Computer Architectures for Machine Perception, 1995. Proceedings. CAMP '95*, 1995, pp. 258-263.

[4] L. Xianru and C. Zixing, "Advanced obstacles detection and tracking by fusing millimeter wave radar and image sensor data," in *Control Automation and Systems (ICCAS), 2010 International Conference on*, 2010, pp. 1115-1120.

[5] W. B. Seales, J. E. Lumpp, and Y. Cheng Jiun, "Model-based tracking in compressed video," in *Aerospace Conference, 1997. Proceedings., IEEE*, 1997, pp. 157-168 vol.1.

[6] M. Piccardi, "Background subtraction techniques: a review," in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, 2004, pp. 3099-3104 vol.4.

[7] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, 2000, pp. 142-149 vol.2.

[8] F. Dellaert and C. Thorpe, "Robust car tracking using Kalman filtering and Bayesian templates," presented at the SPIE Conference on Intelligent Transportation Systems, 1997.

[9] Y. Changjiang, R. Duraiswami, and L. Davis, "Fast multiple object tracking via a hierarchical particle filter," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, 2005, pp. 212-219 Vol. 1.

[10] K. Nummiaro, E. Koller-Meier, and L. Gool, "Object Tracking with an Adaptive Color-Based Particle Filter," in *Pattern Recognition*. vol. 2449, L. Gool, Ed., ed: Springer Berlin Heidelberg, 2002, pp. 353-360.

[11] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *Signal Processing, IEEE Transactions on,* vol. 50, pp. 174-188, 2002.

[12] L. Jiayan, L. Xiaofeng, D. Lianyi, and L. Hengli, "Moving Target Tracking via Particle Filter Based on Color and Contour Features," in *Information Engineering and Computer Science (ICIECS), 2010 2nd International Conference on*, 2010, pp. 1-4.

[13] R. C. Gonzalez, *Digital Image Processing*: Pearson Education, 2009.

[14] J. M. Hammerslay and K. W. Morton, "Poor man's Monte Carlo," *Jounal of the Royal Statistical Society,* vol. 16, pp. 23-28, 1954.

[15] J. E. Handschin and D. Q. Mayne, "Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering†," *International Journal of Control,* vol. 9, pp. 547-559, 1969/05/01 1969.

[16] H. Akashi and H. Kumamoto, "Random sampling approach to state estimation in switching environments," *Automatica,* vol. 13, pp. 429-434, 7// 1977.

[17] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *Radar and Signal Processing, IEE Proceedings F,* vol. 140, pp. 107-113, 1993.

[18] P. M. Djuric and S. J. Goodsill, "Guest editorial special issue on monte carlo methods for statistical signal processing," *Signal Processing, IEEE Transactions on,* vol. 50, pp. 173-173, 2002.

[19] A. Doucet, N. d. Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*: Springer, 2001.

[20] Z. Tao, F. Shumin, L. Xiaodong, and L. Hong, "An Improved Particle Filter for Tracking Color Object," in *Intelligent Computation Technology and Automation (ICICTA), 2008 International Conference on*, 2008, pp. 109-113.

[21] Z. Li-hui, C. Jie, Z. Juan, and D. Li-hua, "The Comparison of Two Typical Corner Detection Algorithms," in *Intelligent Information Technology Application, 2008. IITA '08. Second International Symposium on*, 2008, pp. 211-215.