



Visual Tracking Using Particle Filter Based on Gabor Features

Mohammed Lahraichi^{1*} Khalid Housni¹ Samir Mbarki¹

¹MISC Laboratory, Ibn Tofail University, Faculty of Sciences Kenitra, Morocco

* Corresponding author's Email: lahraichi.mohamed@gmail.com

Abstract: Object tracking is a crucial step in video analysis, as well as detection and recognition. Particle filter, based on color histogram, is considered among multiple approaches that prove their effectiveness in this domain. It is apparent that if an object and its background, or more objects to be tracked, having nearly the same color histogram, the quality of tracking will be negatively affected. Therefore, combining color histogram and texture features for pixels in motion is considered a potential solution to this issue. In this paper, we propose a particle filter tracking method consisting of two stages: at the prediction stage, the state space model is represented by Box–Muller transform, and at the correction stage, the observation likelihood model is calculated in three steps: firstly, the background subtraction is used to extract the moving objects in order to reduce the impact of background pixels on color histogram. Secondly, the texture Gabor features are computed using moving pixels to distinguish between objects having the same color histogram. Finally, the histogram and texture features are combined to define a likelihood observation model. The proposed method is implemented using several video sequences. The visual and numerical results obtained are compared with those of traditional particle filter method, and show that our method is very efficient particularly in case of similar objects, and clearly improves the quality of tracking.

Keywords: Object tracking, Color histogram, Particle filter, Gabor features.

1. Introduction

Visual tracking is an important step in video analysis field. It has broad application prospects in many domains [1, 2]: video surveillance, human-computer interaction and intelligent traffic. However, it still suffers from a lack of robustness due to the variations of illumination, abrupt changes in motion velocity, fast motion, pose variations, small targets, scale changes, cluttered background, and occlusion [3]. Many tracking methods have been developed and evaluated in the last decades to overcome these weaknesses, which can be categorized into three fields:

Point and region tracking: Moving objects are represented by their feature points or region, detected prior or during tracking, and the correspondence between objects across frames is done by establishing a relation between their points or region features. Template matching, probabilistic

methods and optical flow are the famous trackers in this category.

In region-based tracking method, the simplest one is template matching [4] that is if we take a block of an image, we will find the block in the second image which corresponds best to the initial block, by minimizing an error criterion, which is a correlation coefficient defined as a sum of square differences or absolute differences. However, this method is time consuming when image size is large. In addition, if objects have large deformations caused by various poses, or in presence of occlusions, the accuracy of tracking significantly deteriorated. In [5, 6], the authors proposed an improved template matching method, where they have tried to predict the various poses to ensure a good tracking accuracy.

As point tracking method, the optical flow method is described as the displacement of pixels from one image to the next one, which provides an estimate of the real motion of the objects in the scene. Most optical flow estimation methods rely on

a fundamental assumption: Pixel intensity does not change between two successive images. KLT (Kanade–Lucas–Tomasi) is an example of searching good points features and tracking them [7]; it is successfully adopted by estimating that the local velocity is assumed constantly on a spatial neighborhood.

Probabilistic methods aim to predict the estimation according to the system model and correction of this prediction thanks to the new measurements. In the case of Gaussian noise and linear system, kalman filter is considered as a good estimator [8].

Kernel tracking: It uses template density based on appearance and multi-view appearance models, where targets are tracked by computing the motion of the kernels that represent the appearance or the shape of these targets. The kernel can be a rectangular template or an elliptical shape with an associated histogram. Mean shift tracker [9] is a kind of kernel tracking. It is employed to derive the object candidate that is the most similar to a given model by comparing the histogram of the object model in the current frame and the histogram of object candidate in the next frame, until finding maximum similarity between histograms, which are defined as the Bhattacharya coefficient, yet, algorithms based on mean shift may converge to a local maximum, and they are also sensitive to occlusions and objects with quick motion. A large number of improved algorithms [10, 11] based on Mean Shift, such as CAMSHIFT (Continuously Adaptive Mean Shift) that can handle dynamically changing color distribution by adapting the search window size and computing color distribution to a search window, are proposed.

Silhouette tracking: It uses methods like contour evolution and matching shapes methods [3] that given the target model, the object is tracked by estimating the region in each frame. H. Fang et al. in [12] use Active Contour (Snakes) which lock onto nearby edges by minimizing an energy function guided by images and external constraint forces to track multiple objects. Yalmaz et al. [13] propose an active contour model based on level sets that are applied to multiple objects tracking. But in presence of noise, edge blur or occlusion, it uses prior information of color, texture and shape that give exact objects' edges [14].

In the case of Gaussian noise and linear system dynamics, Kalman filters are optimal estimators. So, in the case of non-linear systems with noise, which is not necessarily a Gaussian one, particle filters deal with this kind of issues.

The particle filter algorithm estimates the state of the system by describing the posterior probability distribution of the actual problem through a set of weights. The weight of the particle is calculated by measuring the similarity between the observed image and the target model, thus the choice of robust observation model, which is critical to calculate the weight of particles, is required to effectively track the object with less number of particles. This depends on how the object is modeled.

In this paper, we focus on point tracking in Bayesian inference context, especially the particle filters. We propose a both feature tracker method, where the target representation is based on color histogram and Gabor features to model correctly the observation likelihood.

The remaining part of this paper is organized as follows: the second section presents previous works. The third section describes steps of traditional particle filter method. The fourth section introduces our proposed method that is an amelioration of traditional particle filter method. In the fifth section we introduce the experiments results, and a conclusion in the final section.

2. Related work

Particle filter methods [15, 16] have shown proofs to be efficient in visual tracking. The traditional particle filter method uses color histogram as the main features for modeling the object to be tracked, where the weights of the particles are always the normalized Bhattacharyya distances between the color histogram of object model and the one of regions determined by particles. In [17], the authors have used color-based particle filter for objects tracking, where the global histogram was replaced by a spatial layout with multipart color modelling. Yang D et al. in [18] investigate a new face tracking method which combines an improved spatial histogram with particle filter. In order to improve the quality of tracking, K. Nummiaro et al. have proposed to model the objects by adaptive color distributions, combined with edge-based image features [19]. So as to track multiple objects, B. Sugandi et al. have also implemented the color as a discriminative feature [20]. However, the color histogram degrades the tracking performance in case of occlusion or when the background and the target have similar color distribution. In contrast, models like Haar-like feature, Co-occurrence matrices, and LBP (Local Binary Patterns) distribution [21] have been used to correctly model the objects, Gabor space feature has

also been used for object tracking in conjunction with frame differencing and deformable template modeling [22].

Additionally, many visual tracking methods combine several features, as in [23], Xia et al. have implemented a tracking algorithm which fuses both Haar-like for the structure of the object, and HOG for the edge features. Yet, in [24] an adaptive tracking combined background weight with color-texture histogram based on mean shift algorithm is presented to accomplish precise tracking in complex scenes. In [25] the histogram of improved LBP and color are combined in mean shift tracking. R. Chen et al. in [26] propose a particle filter object tracking based on color and texture information fusion, in which the object model is jointly represented by spatial-weighted color histogram and LBP texture histogram. The LBP feature is not robust on "flat" image regions since it is based on intensity differences. Within flat image regions, the intensity differences are of small magnitude and highly affected by image noise.

P. Wu et al [27] combined color and SIFT(Scale Invariant Feature Transform) point for particle filter object tracking, because the SIFT feature is invariant to illumination, scale and affine motion although the use of SIFT feature may affect the tracking and remain sensitive to background noise interference when objects are smooth.

The afore mentioned methods may track adequately the moving objects when they have relatively different color from the background. However, they may fail to track in the case of similar color, since the particle filter use similarity measure between target and candidate model which may contain the same color as the background.

This paper seeks to tackle this issue by weighting particles using similarity measure based on both color histogram and Gabor filter magnitude of moving objects.

3. Particle filter

Particle filter is a Bayesian inference-based model which uses a Monte Carlo method sample and approximates the probability density function with a set of random samples with associated weights [28].

3.1 Steps of Bayesian recursive filter [29, 31]

Let X_t denote the state of the system at the current time t , and Z_t the observations up to time t . The filtering problem involves the estimation of the

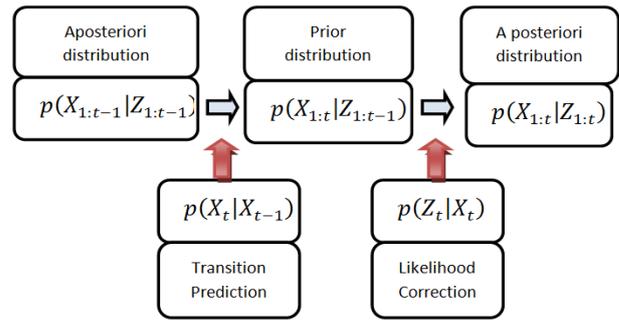


Figure. 1 Steps of recursive Bayesian filter

state vector at time t , given all the measurements (Observations $1:t$) up to and including time t , in a Bayesian setting, this problem can be formulated as the computation of the distribution $p(X_{1:t}|Z_{1:t})$ which can be done recursively in two steps (Fig. 1).

3.1.1. Prediction

In the prediction step, $p(X_{1:t}|Z_{1:t-1})$ is computed from the filtering distribution $p(X_{1:t-1}|Z_{1:t-1})$ at time $t-1$:

$$p(X_{1:t}|Z_{1:t-1}) = \int p(X_t|X_{t-1})p(X_{t-1}|Z_{1:t-1})dX_{t-1} \quad (1)$$

Where $p(X_{1:t-1}|Z_{1:t-1})$ is assumed to be known due to recursion.

The distribution $p(X_{1:t}|Z_{1:t-1})$ can be thought of as a prior distribution across X_t before receiving the most recent measurement Z_t .

3.1.2. Update

The previous prediction is updated using the new measurement Z_t and the Bayes rule, to obtain the posterior across X_t :

$$p(X_{1:t}|Z_{1:t}) \propto p(Z_t|X_t)p(X_t|Z_{1:t-1}) \propto p(Z_t|X_t)p(X_t|X_{t-1})p(X_{1:t-1}|Z_{1:t-1}) \quad (2)$$

In general, the computations in the prediction and update steps cannot be carried out analytically. So the approximation is done using a set of particles, which are projected in the prediction step and sampled by their importance in the update step, allowing the estimate of the new system state. The recursion requires the specification of a dynamic model describing the state evolution $p(X_t|X_{t-1})$, and a model giving the likelihood of any state in the light of the current observation $p(Z_t|X_t)$. The recursion is initialized with some initial distribution $p(X_0)$. When a given descriptor is tracked using a particle filter, the particle weighting is performed

using a similarity measure between the tracked descriptor and each particle descriptor.

3.2 Particle filter

Each particle is represented by: $\{X_t^i, w_t^i\}$ as a possible trajectory of the sequence of states.

X_t^i is the state (in our case is the location) of the particle i at time t while w_t^i is the weight associated with it.

The weight (likelihood particle) represents how important the corresponding particle is, and how similar this particle is to the target model.

The particle filter model is given as follows:

$$p(X_{1:t}|Z_{1:t-1}) \approx \frac{1}{N} \sum_{i=1}^N w_t^i \delta(X_{1:t} - X_{1:t}^i) \quad (3)$$

Where $\delta(\cdot)$ is the Dirac delta function.

First starting with a weighted set of samples $\{X_{t-1}^i, w_{t-1}^i\}_{i=1:N}$ which are approximately distributed according to $p(X_{t-1}|Z_{t-1})$, new samples are generated from a suitable proposal distribution, which may depend on the previous state and the new measurements $X_t^i \sim q(X_t^i|X_{t-1}^i, Z_t)$

The w_t^i is updated each time as follows:

$$w_t^i \propto w_{t-1}^i \frac{p(Z_t|X_t^i)p(X_t^i|X_{t-1}^i)}{q(X_t^i|X_{t-1}^i, Z_t)} \quad (4)$$

The w_t^i is normalized each time and has a unity sum:

$$\sum_{i=1}^N w_t^i = 1 \quad (5)$$

In our case, in order to simplify the tracking system, we have employed the bootstrap particle filter and set:

$$q(X_t^i|X_{t-1}^i, Z_t) = p(X_t^i|X_{t-1}^i) \quad (6)$$

So the Eq. (4) results in:

$$w_t^i \propto w_{t-1}^i p(Z_t|X_t^i) \quad (7)$$

4. Proposed method

In particle filter tracking, the choice of optimal state space and observation models are a crucial step to enhance the quality of particle filter object tracking, since an efficient observation model is required to effectively weight particles.

In this section, we introduce our proposition to calculate observation likelihood model by

combining color features and Gabor filter magnitude, whereas to represent the state space model, we will use Box-Muller transformation.

The flowchart of the proposed method is shown in Fig. 7.

4.1 State space model

In the particle filter algorithm, the prediction step is very essential to predict the state of object in current time based on the previous state, the choice of a model for the state depends on the characteristics of the object to be tracked. There are several models to represent an object such as an ellipse, edges, appearance model, movement information, etc...

In this paper, the object is represented by a rectangle, so the state vector is defined as:

$$X_k = (x_k, y_k, h_k, w_k) \quad (8)$$

Where (x_k, y_k) are the center-coordinates of rectangle, and (h_k, w_k) are the height and width respectively.

The object state $X_{t+1}^i \sim p(X_{t+1}^i|X_t^i)$ is propagated through the second-order autoregressive dynamics model using this formula:

$$X_{t+1} = f(X_t, A) = X_t + A \quad (9)$$

Where A is a random noise, obtained using Box-Muller transform [32] defined as:

$$A = \begin{pmatrix} \sqrt{-2 \ln u_1} \cos(2\pi u_2) \\ \sqrt{-2 \ln u_1} \sin(2\pi u_2) \end{pmatrix} \quad (10)$$

Where u_1 and u_2 are independent random variables that are uniformly distributed in the interval $(0, 1)$.

4.2 Observation likelihood models

4.2.1. Color feature model

The color-based particle filter is used widely for object tracking based on color histogram as a powerful feature to calculate the particle filter weight. Failure of object tracking depends on how this object is distinguished from the background.

So, when the target object and background have similar color distribution, it may be difficult to distinguish the object from background; for example, when we want to track a car that has the same color as the road. To overcome this problem, we have used background subtraction based on mixture Gaussians model [33] and contour extraction to remove background, and detect moving objects (see

Fig. 2(d)), and calculate color histogram using only foreground pixels, represented by white pixels in detected mask (see Fig. 2(c).) as follows :

If (pixel-value in mask == 255) then
Histogram [pixel-value in current frame]++

As shown in Fig. 2. The histogram of moving object (d) is more significant because it hides the background details and represents only the object model compared to that one of the entire image in (b).

The color model is obtained by histogramming techniques in the Hue-Saturation-Value (HSV) color space. In our case, the color information in the HSV space is divided into $m=10 \times 10 + 10$ bins.

The Bhattacharyya distance between the color distribution p of the target candidate and the color distribution q of the target model is given as:

$$d = \sqrt{1 - \rho[p, q]} \quad (11)$$

And Bhanacharyya coefficient ρ is given as:

$$\rho[p, q] = \sqrt{\sum_{u=1}^m p^u q^u} \quad (12)$$

The observation likelihood function based on color histogram is defined as:

$$p_{Color}(Z_t | X_t^i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(1-\rho[p(X_t^i), q])}{2\sigma^2}\right) \quad (13)$$

Where σ^2 is a Gaussian variance.

To overcome the pose and illumination changes of the object to track, the color distribution q of the target model can be updated recursively using adaptive method as follows:

$$q_t = (1 - \alpha)q_{t-1} + \alpha p_t \quad (14)$$

Where p_t is the target object at time t . And $\alpha=0.2$ weighs the contribution of the target object to the target model distribution.

4.2.2. Color feature model

Color histogram is not sufficient to ensure a good tracking based on particle filter, especially if we want to track multiple objects having nearly the same histogram features. So, the idea behind using texture, also, for tracking is that object texture often differs from the background texture as well as from the other objects texture.

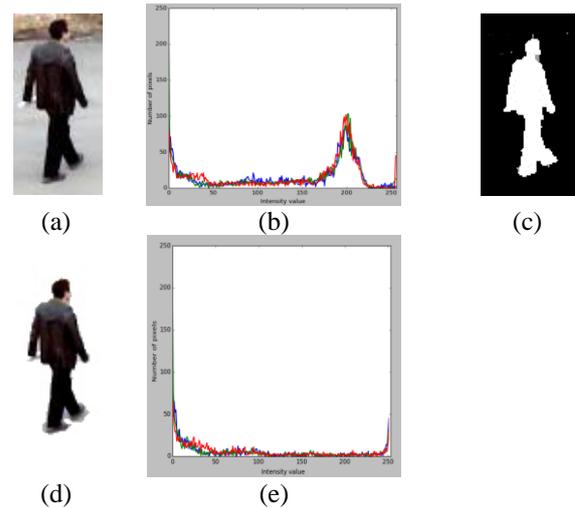


Figure. 2 Object detection using Mixture of Gaussian model and contour: (a) original image, (b) histogram of original image, (c) mask, (d) moving object, and (e) histogram of moving object

Gabor filtering is a widely used image processing, especially to extract texture feature and to imitate human visualization [34]. Gabor filter constitutes a complex sinusoidal plane of particular frequency and orientation modulated by a Gaussian envelope.

The two-dimensional real and imaginary components are represented as follows:

$$G_{Re}(x, y) = \frac{1}{2\pi\sigma^2} \exp\left[-\frac{(x'^2 + y'^2)}{2\sigma^2}\right] \cos\left(2\pi \frac{x'}{\lambda}\right) \quad (15)$$

$$G_{Im}(x, y) = \frac{1}{2\pi\sigma^2} \exp\left[-\frac{(x'^2 + y'^2)}{2\sigma^2}\right] \sin\left(2\pi \frac{x'}{\lambda}\right) \quad (16)$$

Where

$$X' = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} X \quad (17)$$

And $X = (x, y)$, $X' = (x', y')$. λ represents the wavelength of the sinusoidal factor of the Gabor function, θ is the orientation of the filter and σ is the standard deviation of the Gaussian window.

To extract Gabor features, the image is filtered with a bank of filter using various orientations, frequencies and standard deviations. So the design of Gabor filters for features extraction needs an appropriated selection of those filter parameters. Thus Gabor features representation of the image (see Fig. 3) is the result of image $I(x, y)$ convolution with the bank of $n = 12$ Gabor filters (4 orientations, separated by 45 degrees, and 3 frequencies). The convolution result is a complex value which can be decomposed into real and imaginary part:

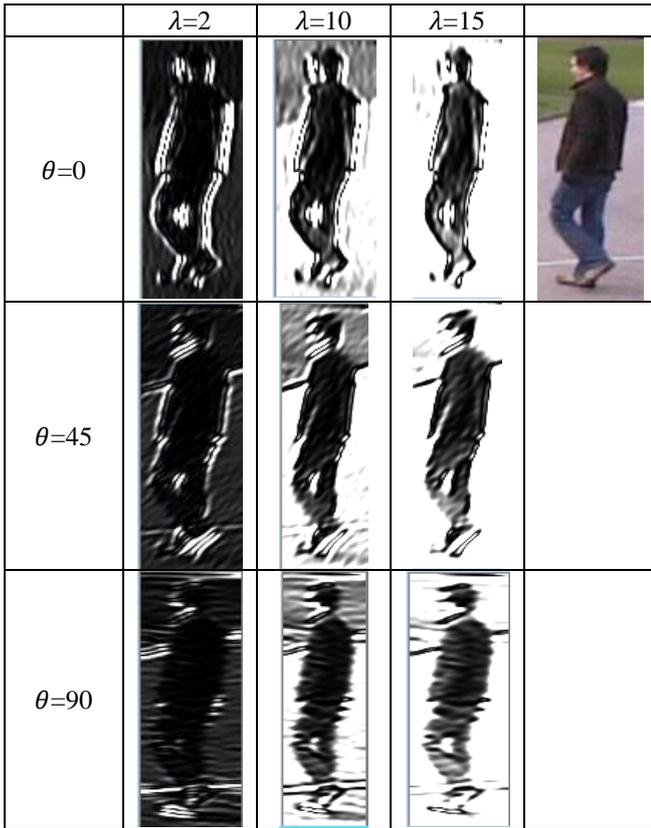


Figure. 3 Example of Gabor filter magnitude obtained by convoluted original image with bank of filter (2 orientations, 3 frequencies, $\sigma=5$ and $\psi=0$)

$$\begin{aligned} I_{Re}(x, y) &= I(x, y) \times G_{Re}(x, y) \\ I_{Im}(x, y) &= I(x, y) \times G_{Im}(x, y) \end{aligned} \quad (18)$$

The phase $\varphi(x, y)$ as well as the magnitude $mag(x, y)$ filter responses can be computed as:

$$\varphi_i(x, y) = \arctan\left(\frac{I_{Im}(x, y)}{I_{Re}(x, y)}\right) \quad (19)$$

$$mag_i(x, y) = \sqrt{I_{Re}(x, y)^2 + I_{Im}(x, y)^2} \quad (20)$$

Where $i = 1 \dots N$.

We define Gabor features as a vector of $n = 12$ elements; each element is the mean of Gabor filter magnitude, because the magnitude is more significant than the phase to represent texture information [35].

$$V = \{v_0, v_1, \dots, v_{n-1}\} \quad (21)$$

$$v_i = \frac{1}{M} \sum_a \sum_b mag_i(a, b) \quad (22)$$

Where M is the total number of the image pixels.

The similarity between Gabor feature target model vector and target candidate is obtained using Euclidean distance :

$$E = \sum_{i=0}^{n-1} |v_i^{model} - v_i^{candidate}| \quad (23)$$

The observation likelihood function based on Gabor filter features is defined as:

$$p_{GF}(Z_t | X_t^i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{E}{2\sigma^2}\right) \quad (24)$$

Where σ^2 is Gaussian variance. To reduce the computational time, we extract the texture features using Gabor filter only for pixels classified as foreground.

4.2.3. Observation likelihood model

In order to combine the color and texture features to represent object model, we represent the observation likelihood function, using color likelihood model $p_{Color}(Z_t | X_t^i)$ and Gabor features model $p_{GF}(Z_t | X_t^i)$ calculated using Eq. (13) and Eq. (24) respectively, by this formula:

$$p(Z_t | X_t^i) = p_{GF}(Z_t | X_t^i) \times p_{Color}(Z_t | X_t^i) \quad (25)$$

And the weight w_t^i is calculated using Eq. (7).

5. Experimental results

In this section, tracking examples are presented to illustrate the benefits of our tracking algorithm, the experiments are implemented using opencv 2.4.9 and c++, as for the number of particles, is set to 100. The proposed algorithm is tested on three real-world video sequences. The first and the second video sequence are from PETS dataset and contain 720×576 pixels per frame, where occlusion occurred in order to confirm the effectiveness of our method. The third video sequence contains scenarios of a person walking and running; the size of frames is 288×448. The objective of this experiment is to demonstrate the performance of the proposed method when the object and background have nearly the same color.

The first video sequence represents some people walking in random directions. The objective is to track a person before, during and after occlusion. The result of the proposed method is shown in Fig. 4. In our experiment, we initially select the person with black jacket from the 151 frame using a green box Fig. 4(a); the target person is tracked correctly in (b) before occlusion, with presence of occlusion in (c), and after occlusion in (d).

The second video sequence shows some people walking randomly; the objective is to track two

people having nearly the same color histogram as shown in Fig. 6.

In order to verify the performance of the proposed method, we have compared our results with those of traditional particle filter method based on color histogram.

Fig.5 presents the results which are obtained based on proposed method in the first row, and those obtained using traditional particle filter in the second row. As an initialization, we select people to track using two boxes (red and green) in frame 340; in (b) the trackers appeared correctly during occlusion and in (c) after occlusion using the proposed method, but the traditional particle filter method does not give a good result during and after occlusion because the trackers have nearly the same color histogram.

In the third video sequence, we select a person in the 198 frame using a box as shown in Fig.8. The results obtained by using the traditional particle filter method are displayed in the first row (green box), and those obtained using the proposed method are displayed in the second row (red box).

In frame 250, the result of proposed method and that of particle filter method show the same performance, though in frame 300 and 315, our proposed method shows more effectiveness compared with particle filter method, because the colors of person and the background are the same.

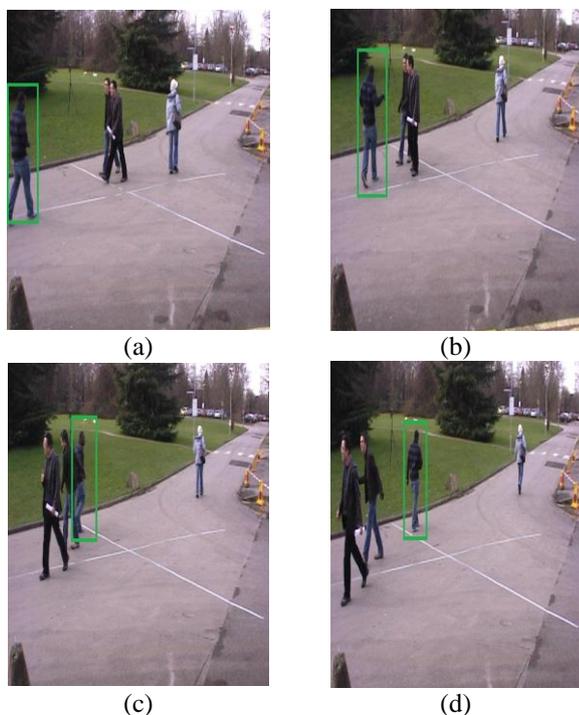


Figure. 4 Results of person tracking using proposed method: (a) first frame 151 (selection of person to track), (b) result of tracking in frame 159, (c) result of tracking in Frame 172, and (d) result of tracking in frame 177

(the person has almost the same color as the trunk of the tree located in the right side of the image).

In addition to verify the effectiveness of the algorithm, we use the distance error between the center of the rectangle of real object (manual acquisition) and the center of the rectangle of the target. The tracking distance errors of the two methods are analyzed and compared.

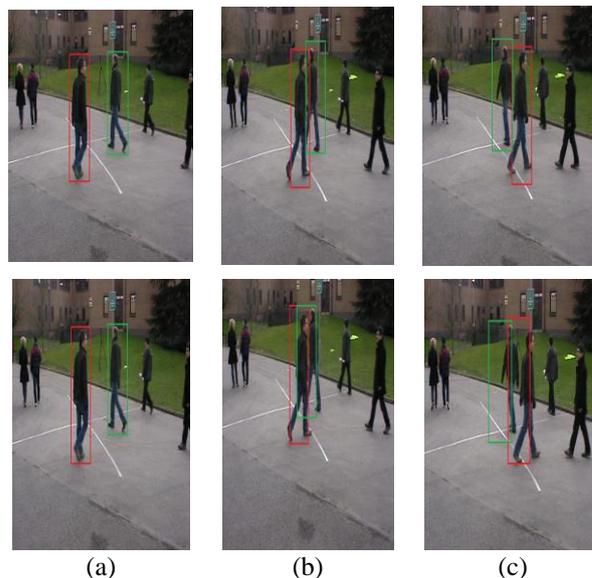


Figure. 5 Results of people tracking using the proposed method (first row) and traditional particle filter (second row): (a) first frame 340 (selection of persons to track), (b) result of tracking in frame 343, and (c) result of tracking in frame 347

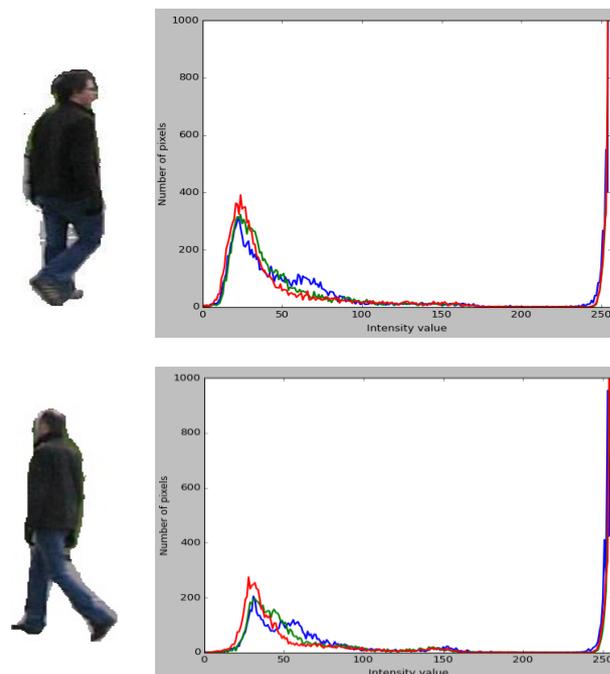


Figure. 6 Color histogram of two people

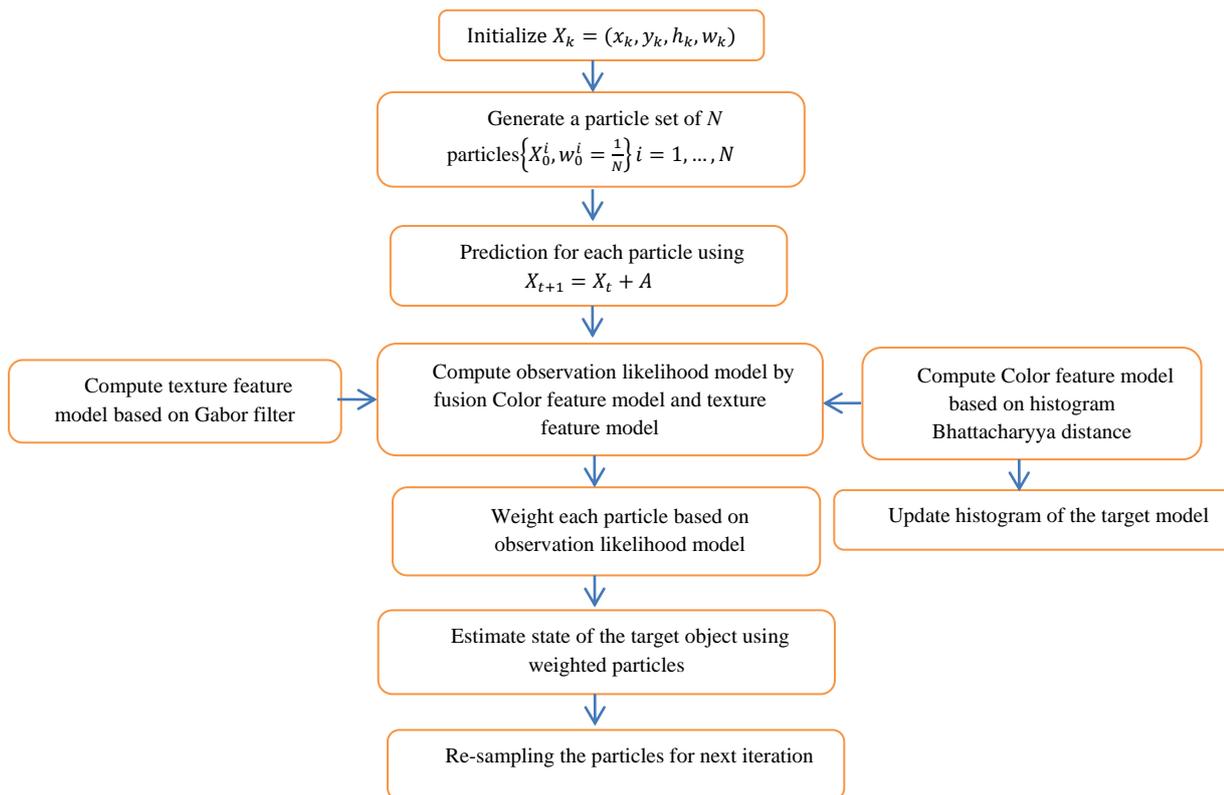


Figure. 7 Flowchart of the proposed method

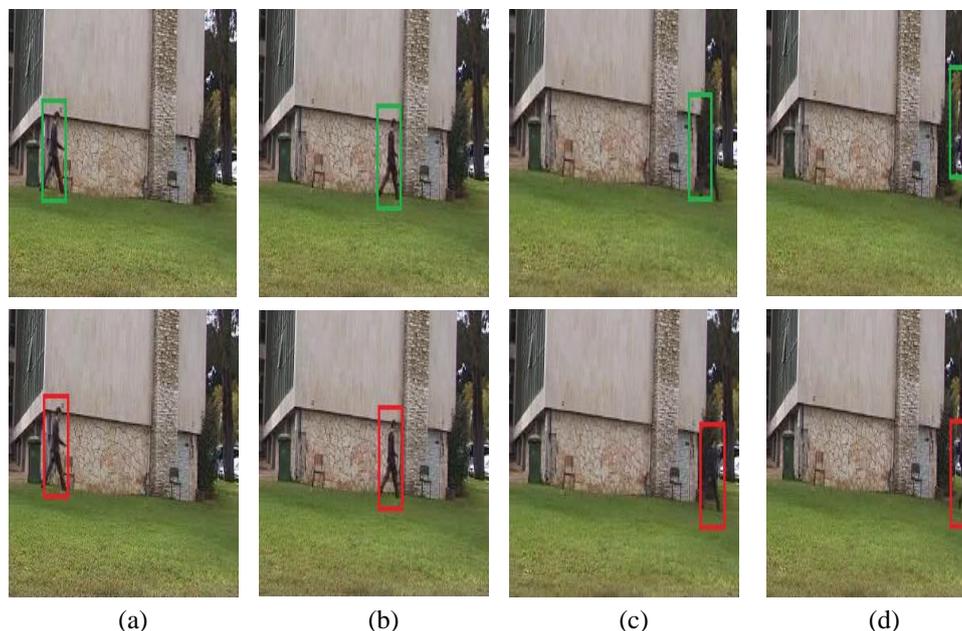


Figure. 8 Results of person tracking using traditional particle filter (first row), and using the proposed method (second row): (a) first frame (frame 198) (selection of persons to track), (b) result of tracking in frame 250, (c) result of tracking in frame 300, and (d) result of tracking in frame 315

Figs. 9 and 10 represent distance error curves of first and third video sequences based on two methods. As can be seen from Fig. 9, when the

environment is relatively simple at the beginning of the tracking, the tracking error of the two algorithms

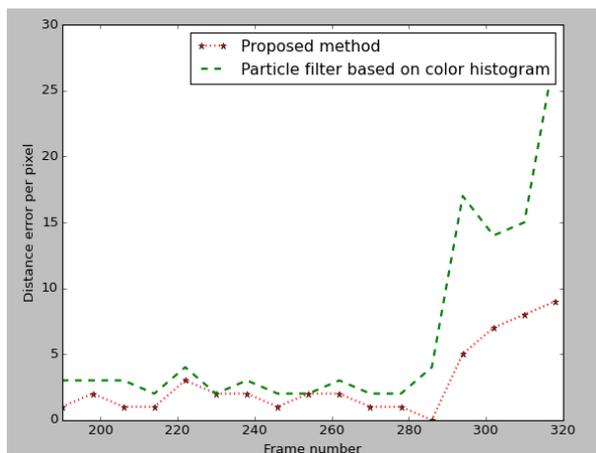


Figure. 9 Distance error curve of third video sequence

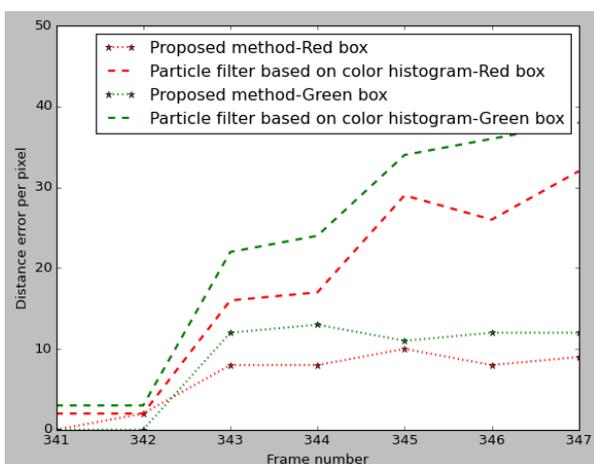


Figure. 10 Distance error curve of second video sequence

is rather small. But after frame 290, when the object has nearly the same color as the background, the error of particle filter algorithm based on color histogram increases obviously, and the method presented in this paper can achieve the goal of stable tracking. In Fig. 10 before occlusion in frame 341 and 342 the tracking error of the two algorithms is nearly the same, but after frame 343 during and after occlusion occurs, our method shows more effectiveness.

6. Conclusion

The object tracking based on color particle filter is sensitive to illumination changes and can be easily disturbed, especially when multiple objects to track have the same color histogram. So, in this paper we have proposed a new method to address this issue, by modeling the observation likelihood based on the fusion of HSV color histogram for foreground pixels obtained from background subtraction with Gabor filter magnitude features, and we have used a state space model based on Box-Muller transform. The results of proposed method have shown object tracking effectiveness compared with traditional

particle filter method, this approach has reduced computational time by calculating the color histogram and Gabor filter features only for foreground pixels.

In a future work, we will focus on object tracking in a camera network, in which we will try to determine which features (invariant signature) can distinguish one object from another across overlapping cameras.

References

- [1] J. J. Deng, X. H. Xie, and B. Doubney, "A bag of words approach to subject specific 3D human pose interaction classification with random decision forests", *International Journal of Graphical Models*, Vol.76, No.3, pp.162-171, 2014.
- [2] L. Calderoni, D. Maio, and S. Rovis, "Deploying a network of smart cameras for traffic monitoring on a city kernel", *International Journal of Experts Systems with Applications*, Vol.41, No.2, pp.502-507, 2014.
- [3] M. Li, C. Wei, Y. Yuan, and Z. Cai, "A Survey of Video Object Tracking", *International Journal of Control and Automation*, Vol.8, No.9, pp.303-312, 2015.
- [4] N. Prabhakar, V. Vaithyanathan, A.P. Sharma, A. Singh, and P. Singhal, "Object tracking using Frame differencing and template matching", *Research Journal of Applied Sciences, Engineering and Technology*, Vol.4, No.24, pp.5497-5501, 2012.
- [5] K. J. Cannons and R. P. Wildes, "The applicability of spatiotemporal oriented energy features to region tracking", *IEEE Transactions, Pattern Analysis and Machine Intelligence*, Vol.36, No.4, pp.784-796, 2014.
- [6] R. He, B. Yang, and N. Sang, "Integral Resion-based covariance tracking with occlusion detection", *International Journal of Multimedia Tools and Applications*, Vol.74, No.6, pp.2157-2178, 2015.
- [7] B.D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision", In: *Proc of the 7th international joint Conf. On Artificial intelligence*, Vol.2, pp.674-679, 1981.
- [8] P. Kalane and P. Loni, "Target Tracking Using Kalman Filter", *International Journal of Science & Technology*, Vol.2 No.2, 2250 – 141X, 2012.
- [9] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking", *IEEE Trans.*

- Pattern Analysis Machine Intelligence*, Vol. 25, No. 5, pp.564–577, 2003.
- [10] O. D. Nouar, G. Ali, and C. Raphael, “Improved object tracking with camshift algorithm” In: *Proc. of IEEE International Conf. On Acoustics, Speech and Signal Processing*, 2006.
- [11] I. Leichter, “Mean shift trackers with cross-bin metrics”, *IEEE Transactions, Pattern Analysis and Machine Intelligence*, Vol. 34, No.4, pp.695-706, 2012.
- [12] H. Fang, S. H. Kim, and J. W. Jang, “A snake algorithm for automatically tracking multiple objects”, In: *Proc of 18th IEEE International Conf. on Image Processing*, 2011.
- [13] A. Yilmaz, X. Li, and M. Shah, “Object contour tracking using level sets”, In: *Proc. of Asian Conf. On Computer Vision*, 2004.
- [14] J. F. Ning, L. Zhang, D. Zhang, and W. Yu, “Joint Registration and Active Contour segmentation for Object Tracking”, *IEEE Transactions, Circuits and Systems for Video Technology*, Vol.23, No.9, pp.1589-1597, 2013.
- [15] Z. Han, Q. Ye, and J. Jiao, “Combined feature evaluation for adaptive visual object tracking”, *International Journal of Computer Vision and Image Understanding*, Vol.115, No.1, pp.69 – 80, 2011.
- [16] W. Lu, K. Okuma, and J. J. Little, “Tracking and recognizing actions of multiple hockey players using the boosted particle filter”, *International Journal of Image and Vision Computing*, Vol.27, No.1/2, pp.189 – 205, 2009.
- [17] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, “Color-based probabilistic tracking”, In: *Proc. of the European Conf. on Computer Visio*, pp.661 – 675, 2002.
- [18] D. Yang, Y. Zhang, R. Ji, Y. Li ,L. Huangfu, and Y. Yang, “An Improved Spatial Histogram and Particle Filter Face Tracking”, *Genetic and Evolutionary Computing. in Advances in Intelligent Systems and Computing book series*, Vol.329, pp.257-267, 2015.
- [19] K. Nummiaroa, E. Koller-Meierb, and L. Van Gool, “An adaptive color based particle filter”, *International Journal of Image and Vision Computing*, Vol.21, No.1, pp.99–110, 2003.
- [20] B. Sugandi, H. Kim, J. K. Tan, and S. Ishikawa, “A color-based particle filter for multiple object tracking in an outdoor environment”, *International Journal of Artificial Life and Robotics*, Vol.15, No.1, pp.41 – 47, 2010.
- [21] M. Heikkila and M. Pietikainen, “A texture-based method for modeling the background and detecting moving objects”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.28, No.4, pp.657 – 662, 2006.
- [22] Y. Wang, J. F. Doherty, and R. E. V. Dyck, “Moving object tracking in video”, In: *Proc. of 29th Applied Imagery Pattern Recognition Workshop*, p.95, 2000.
- [23] C. Xia, S.F. Sun, P. Chen, H. Luo, and F.M. Dong, “Haar-Like and HOG Fusion Based Object Tracking”, In: *Advances in Multimedia Information Processing – PCM 2014. Lecture Notes in Computer Science*, vol.8879, pp.173-182, 2014.
- [24] L. Chen, Q. Huang, L. Pang, and F. Su, “A Robust Tracking Combined with Texture Feature and Background-Weighted Color Histogram”, In: *Proc. of the International Conf. on Communications, Signal Processing, and Systems. Lecture Notes in Electrical Engineering*, Vol.386, pp.751-759, 2016.
- [25] J. Xiaoyan and Q. Shiru, “A target tracking algorithm based on mean shift with feature fusion”, In: *Proc. of 34th Control Conference*, 2015.
- [26] R. Chen, Z. Zhang, H. Lu, H. Cui, and Y. Yan, "Particle-filter-based object tracking with color and texture information fusion", In: *Proc of SPIE - The International Society for Optical Engineering, MIPPR 2009: Automatic Target Recognition and Image Analysis*, 2009.
- [27] P. Wu, L. Kong, F. Zhao, and X. Li, “Particle filter tracking based on color and SIFT features”, In: *Proc. of IEEE International Conf. On Audio, Language and Image*, pp.932-937, 2008.
- [28] S. Kuppuswamy and B. Panchanathan, “Similar Object Detection and Tracking in H.264 Compressed Video Using Modified Local Self Similarity Descriptor and Particle Filtering”, *International Journal of Intelligent Engineering and Systems*, Vol.10, No.5, pp. 95-104, 2017.
- [29] S. Atulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking”, *IEEE Transactions on Signal Processing*, Vol.50, No.2, pp.100-107, 2001.
- [30] A. Y. Carmi, L. Mihaylova, and F. Septier, “Subgradient-based Markov Chain Monte Carlo particle methods for discrete-time nonlinear filtering”, *International journal of Signal Processing*, Vol.120, pp. 532–536, 2016.

- [31] B.N. Vo, M. Mallick, Y. Bar-Shalom, S. Coraluppi, R. Osborne, R. Mahler, and B. Vo, "Multitarget tracking", In: *Wiley Encyclopedia of Electrical and Electronics Engineering*, New York, 2015.
- [32] G. E. P. Box and Mervin E. Muller, "A Note on the Generation of Random Normal Deviates", *The Annals of Mathematical Statistics*, Vol. 29, No. 2, pp. 610–611, 1958.
- [33] C. Stauffer and W.E.L. Grimson, "Adaptive background mixture models for real-time tracking", In: *Proc. of IEEE CVPR 1999*, pp. 24&252, 1999.
- [34] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the Fisher Kernel for Large-Scale Image Classification", In: *Proc of European Conf. On Computer Vision*, pp.143-156, 2010.
- [35] Q. Huo, Y. Ge, and Z.D. Feng, "High performance Chinese OCR based on Gabor features, discriminative feature extraction and model training", In: *Proc. of ICASSP*, Vol.3, p.1, 2001