

AN IMPROVED PARTICLE FILTER TRACKING SYSTEM BASED ON COLOUR AND MOVING EDGE INFORMATION

Chao-Ju Chen¹, Wernhuar Tarng² and Kuo-Hua Lo³

¹Graduate Institute of Computer Science, National Hsinchu University of Education, Taiwan (R.O.C)

²Graduate Institute of e-Learning Technology, National Hsinchu University of Education, Taiwan (R.O.C)

³Industrial Technology Research Institute, Taiwan (R.O.C)

ABSTRACT

A colour-based particle filter can achieve the goal of effective target tracking, but it has some drawbacks when applied in the situations such as: the target and its background with similar colours, occlusion in complex backgrounds, and deformation of the target. To deal with these problems, an improved particle filter tracking system based on colour and moving-edge information is proposed in this study to provide more accurate results in long-term tracking. In this system, the moving-edge information is used to ensure that the target can be enclosed by the bounding box when encountering the problems mentioned above to maintain the correctness of the target model. Using 100 targets in 10 video clips captured indoor and outdoor as the test data, the experimental results show that the proposed system can track the targets effectively to achieve an accuracy rate of 94.6%, higher than that of the colour-based particle filter tracking system proposed by Nummiaro et al. (78.3%) [10]. For the case of occlusion, the former can also achieve an accuracy rate of 91.8%, much higher than that of the latter (67.6%). The experimental results reveal that using the target's moving-edge information can enhance the accuracy and robustness of a particle filter tracking system.

KEYWORDS

Particle Filter, Computer Vision, Target Tracking, Moving Edges, Robustness

1. INTRODUCTION

The popularity of digital cameras and video equipment has changed the human lifestyle and behaviour of event monitoring significantly. In the past time, monitoring systems were mainly installed at some important places, such as street intersections, buildings, parking lots, and so on. Because of the reduction in cost and size as well as the rapid development of computer vision techniques, monitoring equipment has been widely applied in various places, e.g., nursing home, kindergarten and school, for safety supervision. Traditional monitoring is performed using analog cameras operated by security personnel and they usually have to watch the screen all day long. However, according to the survey statistics of Topology Research Institute [1], security personnel would lose concentration and could not react to 95% of screen activities due to visual fatigue after watching multiple monitors for 22 minutes. This is one of the major reasons why traditional human monitoring systems are not reliable.

Today, many analog monitoring cameras are replaced with digital monitoring cameras. The latter can transmit the recorded videos to computers for immediate analysis using computer vision (CV) and image process (IP) techniques. Among them, detection and tracking techniques have been widely applied to many types of security systems, including real-time monitoring systems, intrusion detection systems, and surveillance systems, etc. These systems are able to report sudden incidents immediately to avoid tragedies due to human carelessness or fatigued security personnel. They can also be used to replace lots of boring and time-consuming jobs to reduce the human cost effectively. Therefore, monitoring techniques based on CV have become an active area of research.

In the field of CV monitoring, there have been a lot of research results about detection and tracking techniques applied in various fields. Although many methods have been proposed, some challenges still exist in real and complex environments such as illumination variation, occlusion in complex backgrounds, objects with similar colours, long-term stationary or rapidly moving targets, and so on, leaving more space for researchers to improve their techniques. Target tracking is a very important task for people to do in their daily life, for example, counting the number of passengers, monitoring traffic flow and air traffic control. In recent years, many researchers have engaged in this field of study such that the digital monitoring systems originally applicable to simple environments have now become applicable to real and complex environments to achieve more accurate results.

In the early stages, Haritaoglu et al. [2] proposed a person tracking system based on block comparison. It utilizes the gray-level image of a person as the block feature to search and compare with the foreground region of the newly arrival frame after the person moves, and the objective is to identify the most similar block as the detected location of the person. The foreground detection for such an approach simply computes the average value of all pixels in different blocks, and the foreground region is determined as the location with a large difference in the average value between consecutive frames. El-Azim et al. [3] proposed a block-matching algorithm which finds out the corresponding block of a moving object and then calculates its moving vector to obtain the object's new location.

Stauffer and Grimso [4] proposed another foreground and background detection method for human detection and tracking. Their approach assumes the use of a fixed camera. A background model for the environment is created in advance and the model is updated continuously as time goes by to maintain its correctness. The foreground objects in each frame can be detected by using the background model and their moving trajectories can also be generated. This approach has to create a set of Gaussian mixture models for all pixels as the background model. Therefore, the pixels of the new image for detection are compared with those in the previous model. The reason for doing it is to set a threshold value to find out the most possible background model such that the foreground region can thus be segmented. Some other researchers also engaged in this foreground segmentation approach [5][6].

Heikkila and Pietikainen[7] considered using texture as the feature for tracking, and they replaced the Gaussian mixture model with a multiple-texture histogram such that the target could be located more accurately. Although this approach seems to be very stable according to the final results, there are still some shortages, for example, the cameras have to be fixed and the calculation complexity is very high. Instead of using the matching approaches described above, some researchers utilized the information obtained from a moving object to achieve its detection. Because an object will result in brightness variation during its movement, the moving object can be located by analyzing the difference of pixels between adjacent frames. Such an approach includes the optical flow and temporal spatial methods [8]. However, the results are not desirable for static or slow objects. Hence, these methods have to be combined with other approaches to produce better results for monitoring and tracking applications.

Comaniciu et al. [9] used a mean shift algorithm to track objects. They created a colour histogram for the target first, and then set the core region and the starting point for tracking. The sample pixels in this region were compared with the colour histogram to obtain a similarity score. Then, the score for these pixels was transformed to a weighed value. Also the center of this region was moved to the estimated location. After a certain rounds, the center would converge to the target's actual location gradually. The advantage of this approach is to find the target's location quickly, but the result may be incorrect or unstable as the size of the target changes (for example, the target is moving from far to near). Besides, if the sample pixels at the initial location are not enough or their score is too low, the core region cannot be guided for movement and the object cannot be tracked correctly.

Nummiaro et al. [10] proposed to use colour as the feature for tracking objects. In their approach, a particle filter is developed using the colour histogram as the target model for tracking. When searching and matching the target in a new frame, a certain number of sample pixels are scattered according to the probability density function. The system will create a candidate colour histogram for these sample pixels to match with the target's histogram. Finally, the location of the target is predicted by the score after comparing with these sample pixels. Although the particle filter tracking system based on colour information can achieve effective target tracking, there are still some limitations in practical applications, for example, incorrect results may often occur for the cases of the target and its background having similar colours, occlusion in complex backgrounds, and deformation of objects under tracking.

To solve the problems mentioned above, this study proposes an improved particle-filter tracking system using colour and moving-edge information to enhance the tracking accuracy. It can select a suitable bounding box to enclose the target region when comparing it with the background. The objective is to guarantee the correctness of the target model when updating its information to achieve a better result in long-term tracking. In this study, the performance of the proposed system is evaluated through experiments by comparing it with that of Nummiaro's system.

2. CV TRACKING TECHNIQUES

The CV tracking techniques are divided into (1) creating the background model to detect foreground objects by background matching in a series of images, and (2) creating the target model to detect the foreground objects by foreground matching in a series of images. The common goal of both approaches is to discover the target's location and moving direction continuously. In general, we may use the background subtraction method or manual marking to obtain the pixel set in the target region, followed by selecting appropriate features to describe the target region. For different kinds of objects, the features adopted may also be different. For example, we can use external properties (the boundary and contour) or internal properties (the pixels composing the region) of the target for description.

Basically, the objects for tracking may be divided into rigid bodies and non-rigid bodies. For rigid bodies without easy deformation, e.g., automobiles, it is more appropriate to use external properties as the features for description. In the contrary, if the objects for tracking are non-rigid bodies, e.g., human beings and animals, it is better to use internal properties as features because deformation may occur during their movement. The features often used in current tracking systems include colour, contour, and optical flow. Using colour as the feature has the advantages of easy computation and matching because the pixel values in the target area are available for calculating the colour score, and it is more accurate and robust than contour and template matching. Thus, colour is used as a major feature for the particle filter tracking system proposed in this study. After obtaining the target's information, the subsequent task is to keep on discovering and tracking the target. There are many search algorithms proposed for target

tracking, e.g., the background subtraction method, Kalman filter and the particle filter, and they are described in more detail as follows.

2.1. Background Subtraction Method

The background subtraction method is an intuitive approach often used to detect foreground objects and its principle is very simple. A model for the background is created in advance, followed by subtracting the pixels of the new image from the background image with their absolute values taken to obtain the foreground region (Figure 1). However, this approach is sensitive to illumination variation and thus not suitable for long-term tracking in outdoor environments. To deal with these problems, many researchers have engaged in the related studies in recent years. Stauffer and Grimson created a Gaussian mixture model with the background update mechanism for each pixel such that the system could process slow illumination variation [4]. However, there are still some problems in their approach. For example, the system may make an incorrect prediction if the target is stationary or moves slowly. Because the tracking system has to create a Gaussian mixture model for each pixel and perform matching pixel by pixel, it is very time consuming such that this approach is not adopted in this study.



Figure 1. The background subtraction method (the new image, background image, and foreground object)

2.2. Kalman Filter

The Kalman filter was proposed by Kalman in 1960 [11] and it is a high performance recursive filter to solve discrete signal processing (DSP) problems in the continuous time domain. In recent years, many studies based on Kalman filter have been conducted because of significant advancements in numerical computation. The theory behind Kalman filter is prediction and update. It uses a prediction mechanism to find out the regularity of a dynamic system, by which the possible target location for the next discrete time instant is calculated. Kalman filter is an algorithm with cyclic prediction and update phases during its continuous operation (Figure 2). As for the prediction phase, a dynamic system equation is utilized to estimate the system states for the next time instant, and the difference between the predicted and actual states is calculated, followed by an error correction.

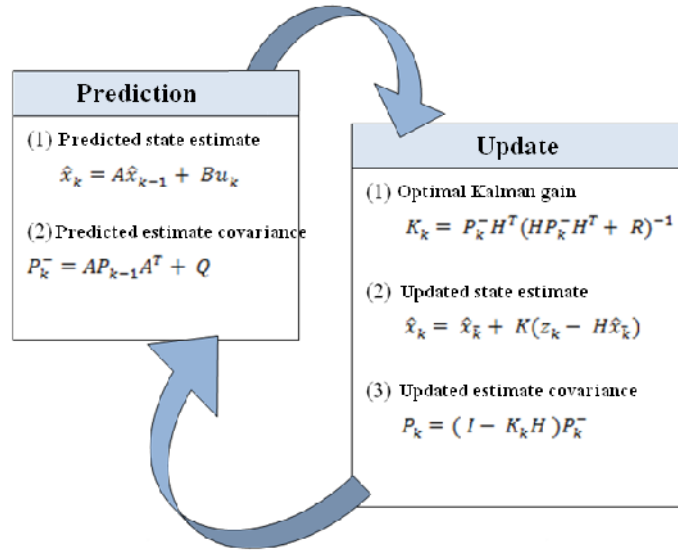


Figure 2. The continuously cyclic prediction and update phases

2.3. Particle Filter

A particle filter is an algorithm applicable to nonlinear, non-Gaussian constraints in response to the limitation of Kalman Filter. Gaussian distribution is often used as the probability density function for linear, Gaussian random processes under Bayes' theorem [12], while Kalman filter is capable of solving such estimation questions effectively. The Extended Kalman filter is the representative solution to the estimation problems for a nonlinear system. However, its model has to be linearized and the probability density function must satisfy Gaussian distribution, but the model is usually nonlinear, non-Gaussian in practical applications. As a result, it is difficult to implement the optimal estimation and therefore the particle filter was proposed for applications in nonlinear systems with non-Gaussian constraints. It may obtain the mathematical expectation of an arbitrary function by using a certain number of sample particles to represent the system's posterior probability distribution. In the studies of CV tracking systems, Sequential Monte Carlo methods similar to that in [13] can be used to implement a particle filter, which contains the following four parts:

(1) Probability Distribution: It is used to define the random state of an object. If the accurate state information is unknown, the probability function is used to describe the state information of an object (Figure 3).

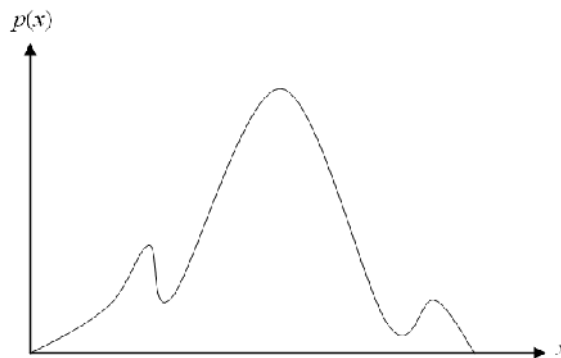


Figure 3. Using the probability density function to describe the state information

(2) Dynamic Model: When the state information of an object changes as time goes by, its probability density function changes as well. The prediction is based on the discrete-time calculation. However, the dynamic model uses a random differential equation for updating its state in real time. The equation can be determined based on the system requirement and its power is allowed to be any order. The probability density function $p(x_t)$ is determined according to the prior probability distribution $p(x_{t-1})$ and thus can be represented as $p(x_t | x_{t-1})$.

(3) Measurement: If Z_t is the measurement for each time step t , where $Z_t = \{z_0, \dots, z_t\}$, $p(x_t)$ is the probability function, $p(x_t | Z_{t-1})$ is the prior probability, and $p(x_t | Z_t)$ is the posterior probability. Given the dynamic model, the posterior probability $p(x_t | Z_{t-1})$ is obtained by the result of the previous time step as:

$$p(x_t | Z_{t-1}) = \int_{x_{t-1}} p(x_t | x_{t-1}) p(x_{t-1} | Z_{t-1}) \quad (1)$$

In a complete tracking procedure, the dynamic model is utilized to calculate the prior probability function $p(x_t | Z_{t-1})$, followed by estimating the posterior probability function $p(x_t | Z_t)$ to provide a new estimation.

(4) Sampling: The posterior probability function $p(x_t | Z_t)$ is usually too complex to be simplified as a simple equation and x is often multi dimensional, so we cannot sample within the interval of $p(x_t | Z_t)$. Therefore, the skill of factor sampling has to be used, that is, a similar probability density function $f(x) = f_2(x)f_1(x)$, $x \in X$ is used for replacement. If the sample set $S = \{s^{(1)}, \dots, s^{(n)}\}$, $s^{(j)} \in X$, $1 \leq j \leq n$, then S will obtain the sample from $f_1(x)$ randomly, while the weight of each sample $s^{(j)}$ is derived as following:

$$\pi^{(j)} = \frac{f_z(s^{(j)})}{\sum_1^N f_z(s^{(j)})}, j = \{1, \dots, n\} \quad (2)$$

Finally, a new sample set may be calculated according to the sample set S . The particle filter shows desirable properties in nonlinear, non-Gaussian systems, so it has been applied in many fields, for example, predicting economic data, tracking flying objects with radar, and monitoring vehicles on the highway.

Although the particle filter has been applied widely in a variety of areas, it still has some shortages. Among them, the major problem is that it needs lots of random samples to approximate the posterior probability distribution. When operating in a real environment, the computation complexity is higher as the environment becomes more complex. Thus, the key point for improving the system performance is to reduce the number of samples. On the other hand, if colour is used as the feature in a particle filter, incorrect prediction may occur easily when the target and the background have similar colours. In this study, a recursive algorithm is used to remedy the drawback of requiring lots of samples. Besides, the algorithm combines the colour and moving-edge information to deal with the situation of incorrect decision due to the target and background with similar colours, occlusion in complex backgrounds, and deformation of objects.

3. SYSTEM DESIGN

In this study, a CV tracking system is created by using the colour and moving-edge information to improve the particle-filter tracking system proposed by Nummiaro et al. to achieve more accurate results. In our system, the target model for tracking has to be created in advance, followed by

comparing it with the foreground region of the new frame to locate the target's position, while the system keeps on updating the target model to remain its accuracy. The major steps of tracking the target by our system are described in the following.

3.1. Defining Target Model

Before tracking, the target region has to be set manually to provide its colour information. After that, the target region is compared with the foreground region in each newly arrival frame to locate the target's position (Figure 4). Determining the target region is very important because it will affect the system accuracy.

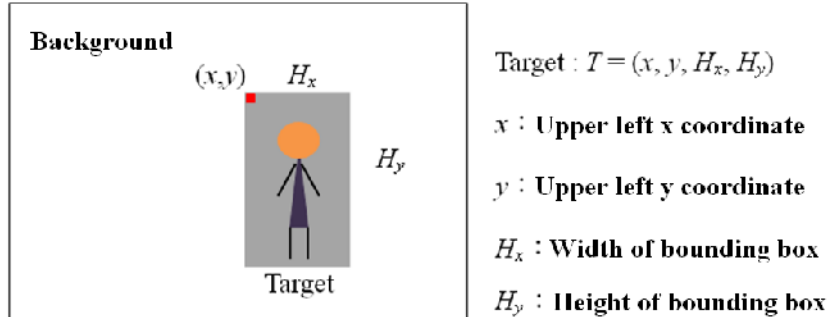


Figure 4. Setting the target region before tracking

3.2. Target Matching Approach

After setting the target region, the proposed system uses colour as the major feature to create the histogram based on the RGB colour space to describe the target's information. To solve the problem of incorrect prediction due to the target and backgrounds with similar colours, the moving-edge information is adopted to improve the system's accuracy. The creation of colour histograms, detection of moving edges and computation of similarity are described in the following.

(1) Colour Histogram

In the proposed system, the RGB colour space is used to create the histogram for the target model (Figure 5). If N represents bin size of the histogram, we have $N = N_r \times N_g \times N_b$. In this study, $N_r = 8$, $N_g = 8$ and $N_b = 8$, so N is computed as 512. At first, the colour histograms, including the RGB values, for each pixel in the target region are created, respectively, and then they are transformed into histograms with bin=8. Furthermore, the three-dimensional histogram is transformed to a one-dimensional histogram with bin= $((\text{bin} \times R) + G) + B$ for easy matching and calculation.

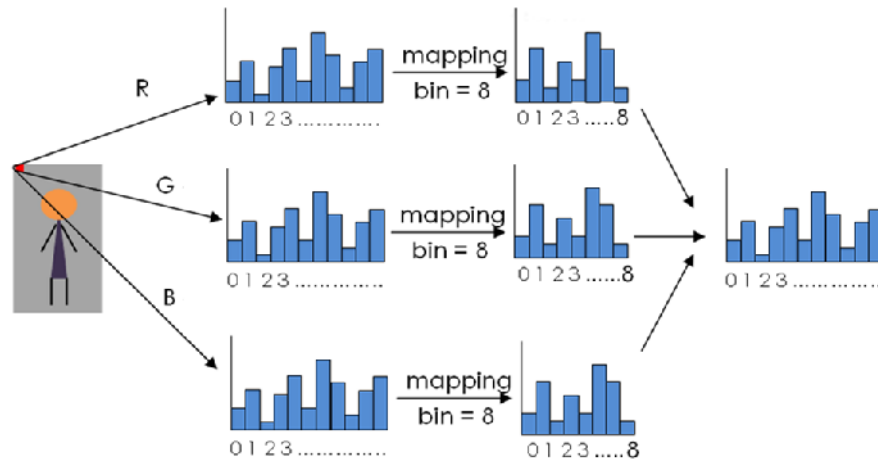


Figure 5. Steps to create the colour histogram for the target model

(2) Moving Edges

The moving edges of the target in a frame can be obtained by subtracting the images of consecutive frames and taking the absolute values of the pixels, followed by retaining those with values higher than the threshold (Figure 6).



Figure 6. Obtaining the moving edges by subtracting consecutive images

(3) Similarity Calculation

When performing matching the target model with the newly arrival frame, three scores are calculated, i.e., colour score, motion score and similarity score. After the system obtains the first two scores, the similarity score can also be calculated. Finally, the system will determine if the target is located or not by checking if the similarity score exceeds the threshold value.

(a) Colour Score

After creating the colour histogram of the target region, the target and the background are compared to find out the location of the target in the background image (Figure 7).

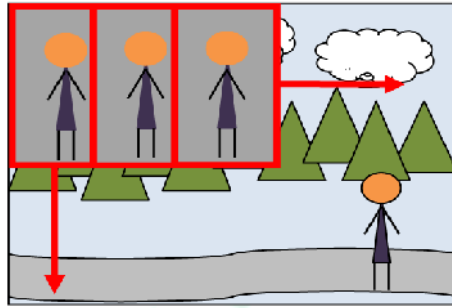


Figure 7. Comparing the colour histograms of the target region and background

In this study, the formula of Bhattacharyya [15] is used for comparison:

$$\rho[p, q] = \int \sqrt{p(u)q(u)} du \quad (3)$$

It is used to calculate the similarity of bins between colour histograms of the target and background. In Equation (3), p and q are the colour histograms of the background and the target, respectively, and u is the number of bins. In the proposed system, the discrete density between bins is considered. Therefore, the approach by Nummiaro et al. [10] is adopted and the integral is replaced by accumulation as following:

$$\rho[p, q] = \sum_{u=1}^m \sqrt{p(u)q(u)} \quad (4)$$

(b) Motion Score

As we obtain the moving-edge information of the target, we can see that the moving edges appear mostly on the moving objects (Figure 8). We first obtain the RGB values of the original image for all pixels on the moving edges, and calculate the corresponding bin value after transformation. After that, we check if the bin value is higher than a certain threshold value in the target model, and keep it if the answer is yes or remove the pixel otherwise.

$$I_d'(x, y) = \begin{cases} 1, & \text{if } q(k) \geq THv \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

In Equation (5), I_d' is the pixel value after filtering, and (x, y) indicates the coordinate of a pixel on the moving edge.

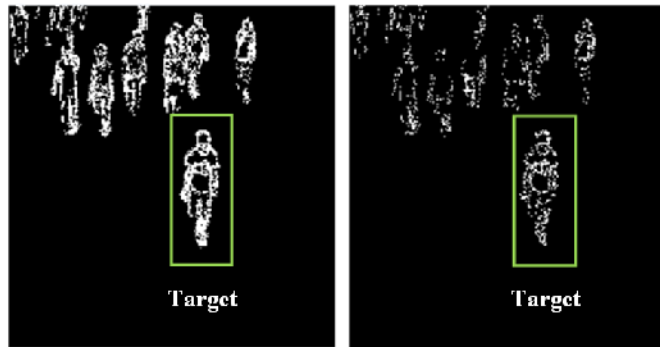


Figure 8. The moving-edge information before (left) and after filtering (right)

After performing the above calculation, a lot of information on non-target edges may be filtered out. However, considering only the moving-edge information is not enough when tracking the target. Because the target under tracking is the person walking in front of the others, the tracking result is deviated upwards if only the moving-edge information is considered since there are also a lot of moving edges detected behind. To avoid such a problem, we design a moving edge filtering mechanism by using the colour information in the target model to filter out the information of non-target edges. After obtaining the filtered image, an equation to compute the similarity of filtered moving edges with the target's moving edges is used in the proposed system, which is defined as the motion score:

$$S_{motion} = C \left(\frac{Area(B_{t_i})}{Area(B_{t-1})} \right) \left[\frac{1}{Area(B_{t_i})} \sum_{(x,y) \in B_{t_i}} I_d'(x,y) \right], \quad C(x) = \begin{cases} 1, & \text{if } x \geq 1 \\ x, & \text{otherwise} \end{cases} \quad (6)$$

In Equation (6), B_{t_i} is the bounding box of the i -th sample particle for the current frame, B_{t-1} is the bounding box of the target model for the previous frame, Area function calculates the area of the bounding box, and (x, y) is the coordinate of the bounding box. The moving-edge information of all sample pixels obtained from Equation (6) must be normalized. When the system determines the target region during the tracking process, its size may be reduced gradually to cause tracking errors if the target region contains very few colours, so the area of bounding box is used as an adjusting factor in Equation (6). The objective is to prevent the bounding box from becoming smaller and smaller; otherwise, it may not be able to cover the entire region of a moving object when determining the target's location.

(c) Similarity Score

After obtaining the colour score and dynamic score, the system transforms them into the distance as following:

$$d = \sqrt{1 - \rho[p, q] S_{motion}} \quad (7)$$

Then, the distance is further converted into the weighted value of a sample particle such that the one with a closer distance will have a larger weighted value. Finally, the weighted value is used as the similarity score in the proposed system. The calculation of the similarity score is shown by the following equation:

$$\pi = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{d^2}{2\sigma^2}} \quad (8)$$

The major reason for transforming the distance value into the weighted value is to enlarge the difference between scores. In Figure 9, the darker colour indicates the lower weight, while the lighter colour indicates the higher weight.

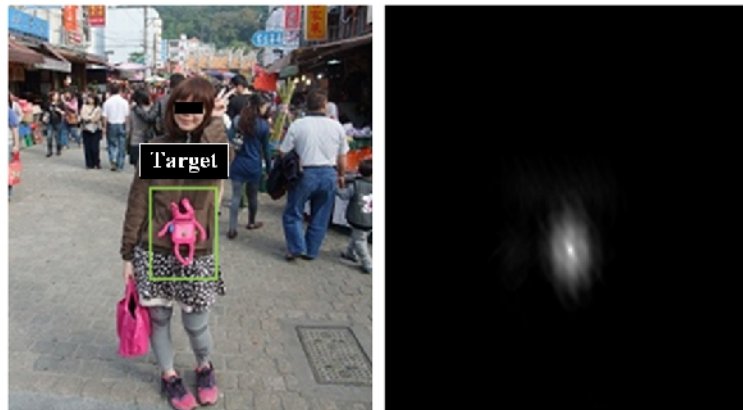


Figure 9. Calculating the weighted values for the target region under tracking

3.3 Improved Particle Filter

In general, the target's position on the image can be located by a simple and accurate way if all pixels of the target and the background are compared one by one. However, the computation workload is very heavy and the speed could be very low such that it is difficult for the system to be applied practically. The particle filter performs well in solving this kind of problems because it utilizes a certain number of random sample particles to represent the posterior probability distribution, so it is applicable to arbitrary non-linear random systems. Nevertheless, the particle filter still has the problem of requiring a lot of sample pixels. In this study, the particle filter tracking system uses a recursive algorithm (similar to that proposed in [14]) to adjust the amount of sample pixels dynamically (Figure 10). At first, the system scatters n sample pixels randomly on the newly arrival image (n is determined by the system), and then the operating procedure is described frame by frame as following:

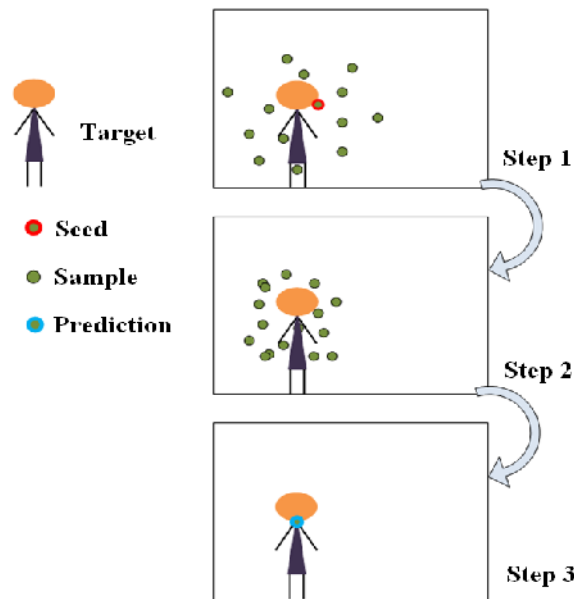


Figure 10. Operation procedure of the improved particle filter tracking system

Frame 1: Set the initial coordinate of the target manually.

Frame 2: Start matching the target region with the background, and continue the following steps:

Step 1: According to the coordinate of the target determined in the previous frame, scatter n sample particles around it randomly. If the score of the sample set does not exceed the threshold value, then the one with the highest score is selected as the seed to form the sample set.

Step 2: Using the seed obtained from Step 1, n samples are scattered randomly to generate a new sample set, followed by checking if the score of the new sample set exceeds the threshold value. If yes, then proceed with Step 3; otherwise, return to Step 1.

Step 3: If the score of the sample set is higher than the threshold value or the process has reached the maximal number of rounds, then the prediction is determined by the current similarity score.

In the proposed system, some random noise perturbation is added to the seed's state to generate new sample sets randomly. The purpose is for the bounding box to bounce randomly such that it can be used to predict the dynamic model of the target. It is assumed that the state vector of some seeds X_k are $[x, y, H_x, H_y]$, where k is the time instant of the frame, (x, y) indicates the coordinate of the seed's bounding box, H_x and H_y are the width and height of the bounding box, respectively. Then, we generate the new sample state by adding the sample vector X_k of a certain seed with the random noise perturbation. By setting a reasonable σ_x for each sample state x , the new state is computed as:

$$x' = x + \Delta x, \quad \Delta x \sim N(0, \sigma_x) \quad (9)$$

When matching the target region with the background, the system performs the above procedure for each element in the random sample set which is generated recursively to obtain the similarity score. If we select the sample with the maximum value as the final prediction result, a prediction error might still occur if the similarity values of all samples are too low. Therefore, we add up the similarity values of all samples in the sample set to see if the threshold value is exceeded. If yes, then the averaged accumulation value is used as the prediction result in this round:

$$E[S_t] = \sum_{n=1}^N \pi_t^{(n)} S_t^{(n)} \quad (10)$$

Otherwise, some samples with higher scores in the sample set are selected as the seeds, followed by executing the next round of to scatter sample pixels randomly. In Equation (10), $\sum_{n=1}^N \pi_t^{(n)} S_t^{(n)} = 1$, N is the number of samples, π is the weighted value, and $E[S_t]$ is the state estimated at time instant t .

The major purpose of the proposed system is to select a small part of better samples as the seed for the next round in the recursive sampling process. Therefore, the number of samples needed is not as large as that in the traditional particle filter for the first random scattering. Instead, the samples with higher scores are selected as the seed for the next round. By doing so, the generated sample will become closer and closer to the target. Furthermore, we set the threshold values in some steps as the terminating condition. If the predetermined recursion count has not been reached while the similarity score is higher than the threshold value in the matching process, then there will be no further recursive rounds executed. Consequently, the number of samples may be reduced without affecting the tracking accuracy.

3.4. Target Update

After using the improved particle filter to match and determine the location of the target, the system has to update the target model if the similarity score is higher than the threshold value before entering the next frame. In the following equation, u is the bin value, $p_{E[S,t]}$ is the weighted value scaled according to score by the system, $p_{E[S,t]}^{(u)}$ is the colour histogram and state of the prediction result after all samples are accumulated and averaged in a certain round.

$$q_t^{(u)} = (1 - \alpha)q_{t-1}^{(u)} + \alpha p_{E[S,t]}^{(u)} \quad (11)$$

The procedure for target tracking using the proposed system is shown in Figure 11. At first, the input video is converted into a number of bmp images and the target model is created by manual selection. After that, the improved particle filter is used for tracking by matching with the background image continuously in the newly arrival frame and updating the target model.

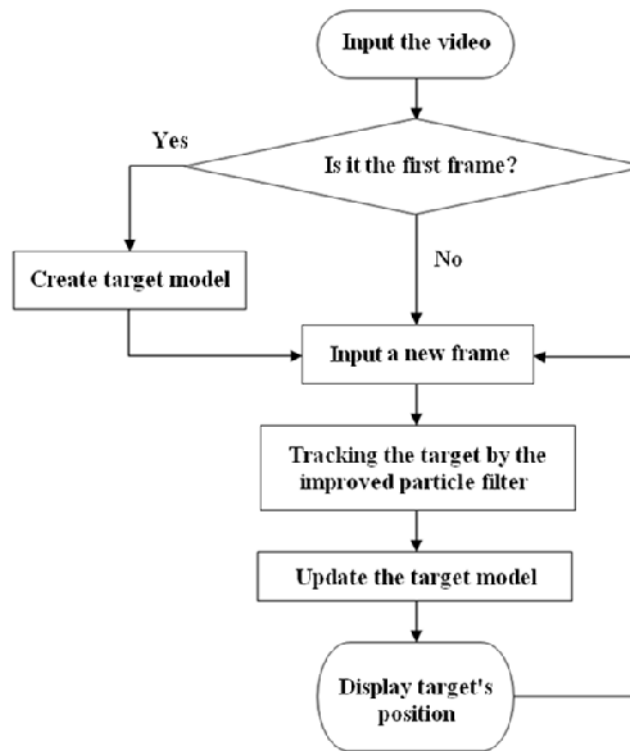


Figure 11. Procedure of target tracking using the proposed particle filter tracking system

4. PERFORMANCE EVALUATION

In this study, the video clips taken in different environments are used to evaluate the performance of the proposed improved particle filter tracking system. The same tests are also performed on the system proposed by Nummiaro et al., and the experimental results are compared by statistical analysis. The software and hardware used in the experiment are listed as following:

- Operating system: Microsoft Windows 7
- CPU: Intel i7-2640M(2.8GHz)
- Memory: 8 GB
- Software: OPEN CV, Microsoft Visual C++ 6.0

4.1. Experimental Video Clips

In this study, 6 types of video clips with a total 100 targets are used to evaluate the tracking performance on both systems. They are: (1) outdoor people, (2) walking through corridor, (3) soccer games, (4) basketball games, (5) traffic flows, and (6) leisure activities. Our main purpose is to test the accuracy and robustness of these two tracking systems under daily life situations.

(1) Outdoor People: the video is taken outdoors randomly to show people walking by. The experiment tracks 9 people continuously for 100 to 300 frames to test the system's tracking ability under the situation when people are moving from far to near with their sizes changing from small to large (Figure 12).



Figure 12. Video 1: outdoor people walking by

(2) Walking Through Corridor: the video is downloaded from the Internet for experimental usage. The experiment tracks 5 people continuously for 270 to 420 frames to test the system's tracking ability under the situation when people are wearing clothes with similar colours and switching their positions through the corridor with illumination variation caused by sunlight and shadows (Figure 13).

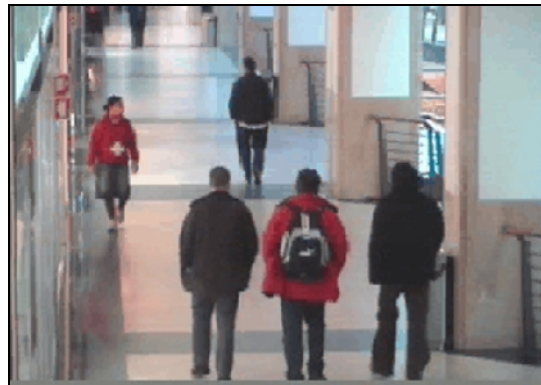


Figure 13. Video 2: people walking through corridor

(3) Soccer Games: (a) the video is a soccer game broadcasted on Azio TV in Taiwan (Figure 14). The experiment tracks the soccer players continuously for 100 to 164 frames; (b) the video is a soccer game broadcasted on now Sports in Hong Kong (Figure 15). The experiment tracks the soccer players continuously for 100 to 330 frames. We want to test the system's tracking ability

under the situation when the target and several other soccer players wearing clothes with the same colours are moving quickly on the sports field.



Figure 14. Video 3(a): a soccer game broadcasted on Azio TV



Figure 15. Video 3(b): a soccer game broadcasted on now Sports

(4) Basketball Games: (a) the video is a basketball game broadcasted on Azio TV in Taiwan (Figure 16). The experiment tracks the basketball players continuously for 100 to 235 frames; (b) the video is a basketball game downloaded from YouTube (Figure 17). The experiment tracks the basketball players continuously for 100 to 206 frames; (c) the video is another basketball game downloaded from YouTube (Figure 18). The experiment tracks the basketball players continuously for 100 to 247 frames. Using these basketball game videos, we want to test the system's tracking ability under the situation when the basketball players intersect with their teammates (wearing the same colour) on the field (moving quickly) and on the bench (moving slowly) and change their posture (deformation).



Figure 16. Video 4(a): a basketball game broadcasted on Azio TV



Figure 17. Video 4(b): a basketball game broadcasted on YouTube



Figure 18. Video 4(c): a basketball game broadcasted on YouTube

(5) Heavy Traffic Flow: the video is downloaded from YouTube and it shows the heavy traffic flow on the street (Figure 19). The experiment tracks the cars and motorcycles continuously for 130 to 520 frames to test the system's tracking ability for moving and turning vehicles with occlusion in a complex environment.



Figure 19. Video 5: the heavy traffic flow on the street

(6) Student Activities: (a) the video shows a performance activity of some elementary school students (Figure 20). The experiment tracks 5 students continuously for 200 to 609 frames; (b) the video shows a coach guiding students in a skating activity (Figure 21). The experiment tracks the students continuously for 115 to 390 Frames. Using the above 2 videos, we can test the tracking ability of the system under the situations when people may move, intersect, crouch, jump and stand still in an indoor environment.



Figure 20. Video 6(a): a performance activity of some elementary school students



Figure 21. Video 6(b): a coach guiding students in a skating activity

For the above 6 types of videos, we list the important factors affecting the accuracy of a tracking system, i.e., the target's moving speed, environmental illumination variation, occlusion in complex backgrounds, and similarity with background colours (Table 1), where the degrees of affection are rated by strong, middle and weak.

Table 1 Factors affecting tracking accuracy in the experimental videos.

Factors affecting tracking	Video 1	Video 2	Video 3	Video 4	Video 5	Video 6
Target's moving speed	weak	weak	strong	strong	strong	middle
Environmental illumination	weak	strong	middle	middle	middle	middle
Occlusion in backgrounds	weak	middle	strong	strong	weak	strong
Similarity with background	weak	weak	middle	strong	weak	weak

4.2. Experimental Results

In this study, the proposed system is tested by 6 types of videos with a total of 10 video clips and 100 targets, and its accuracy rate is 94.6%. The same test is performed on the system proposed by Nummiaro et al, and the accuracy rate is 78.3%. Therefore, the accuracy rate is increased by 16.3% when the moving-edge information is added as a feature. A further investigation shows that tracking errors on their system often occur under the following situations:

- Deformation of objects or illumination variation in backgrounds
- The target intersects with objects having the same (or similar) colours.
- The target and the background have the same (or similar) colours.

When the above situations occur, the bounding box of Nummiaro's system may shrink continuously as time goes by, which is caused by using colour as the only feature for tracking. In that case, a smaller bounding box gives a higher score in the matching process such that the system tends to select a smaller bounding box for the target region when updating the target model. However, the target information provided by the bounding box is also reduced accordingly, and that's the reason causing the tracking error in the long run. In our approach, the colour and moving-edge information are combined to improve the performance of a particle filter. Hence, the bounding box of the target will not have the shrinking problem under situations of environmental illumination variation and deformation of objects and thus better tracking results can be obtained.

The proposed system utilizes the moving-edge information as a major feature to deal with the situation that the target intersects with other objects, so the tracking accuracy can be enhanced significantly. Its robustness under the situation of occlusion in complex backgrounds is further investigated as follows. The statistic results of tracking the target with and without occlusion are provided in Table 2. Using our system, the accuracy rate is 91.8% for tracking the target with occlusion, and is 97.5% for the case without occlusion. For the system proposed by Nummiaro et al., the accuracy rate is 67.6% for the case with occlusion, and is 89.5% for the case without occlusion. The significant difference shows the robustness by using the moving-edge information as the feature for target tracking.

Table 2. Comparing the accuracy rate of our system with that of Nummiaro’s system.

Target	Number of Frames	Our System	Nummiaro’s System
With Occlusion	9613	8826(91.8%)	6497(67.6%)
Without Occlusion	9171	8938(97.5%)	8208(89.5%)
Total	18784	17764 (94.6%)	14705(78.3%)

4.3. Case Study

The following video clip has 390 frames showing a skater on the skating rink intersecting with some other skaters (Figure 22). The factors of the target’s moving speed, environmental illumination variation, occlusion in complex backgrounds and similarity with background colours are rated from middle to strong, respectively. According to the results, our system can locate the target correctly through all frames, while Nummiaro’s system misses the target in the 50th frame due to occlusion. Even though the target appears again after a few frames, Nummiaro’s system still cannot locate it because its bounding box is not selected correctly.

Frame 1:



Frame 50:



Frame 100:



Figure 22. Tracking results with occlusion by our system (left) and Nummiaro’s system (right)

5. CONCLUSIONS

An improved particle filter tracking system based on colour and moving-edge information has been proposed in this study to achieve a higher accuracy rate in long-term tracking. It can select a

more suitable bounding box to match with the target size when updating the target model. In addition, the background subtraction method is utilized to obtain the moving-edge information of the target. The system is effective for the case of occlusion in complex backgrounds. An experiment has been conducted to evaluate the performance of the proposed system by using 6 types of video clips with a total 100 targets to test the accuracy and robustness of the proposed system in daily life situations. According to the experimental results, the accuracy rate of the proposed system (94.6%) is higher than that of the system proposed by Nummiaro et al. (67.6%). For the case with occlusion in complex backgrounds, the accuracy rate of the proposed system (91.8%) is much higher than that of the system proposed by Nummiaro et al. (67.6%). The experimental results reveal that using the moving-edge information in addition to the colour information can further increase the accuracy and robustness of a particle-filter tracking system.

REFERENCES

- [1] Topology Research Institute (2008) *The new trend of development in security monitoring from the viewpoint of intelligent analysis*, <http://www.topology.com.tw>.
- [2] Haritaoglu, I., Harwood D. & Davis, L. S. (1998) "A real time system for detecting and tracking people", *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Santa Barbara, California, June 23-25, 1998.
- [3] El-Azim, S. A., Ismail, I. & El-Latiff, H. A. (2002) "An efficient object tracking technique using block-matching algorithm", *Proceedings of the 19th National, Radio Science Conference*, Alexandria, Egypt, March 19-21, 2002.
- [4] Stauffer, Chris. & Grimson, W. E. L. (1999) "Adaptive background mixture models for real-time tracking", *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Fort Collins, Colorado, June 23-25, 1999.
- [5] Horprasert, T., Harwood, D. & Davis, L. S. (1999) "A statistical approach for real-time robust background subtraction and shadow detection", *Proceedings of IEEE ICCV'99 FRAME-RATE Workshop*, Kerkyra, Greece, September 3, 1999.
- [6] Lim, J. & Kriegman, D. (2004) "Tracking humans using prior and learned representations of shape and appearance", *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition*, Seoul, Korea, May 17-19, 2004.
- [7] Heikkila, M. & Pietikainen, M. (2006) "A texture-based method for modelling the background and detecting moving objects", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 4, pp. 657-662.
- [8] Talukder, A. & Matthies, L. (2004) "Real-time detection of moving objects from moving vehicles using dense stereo and optical flow", *Proceedings of IEEE/ESJ International Conference on Intelligent Robots and Systems*, 28 September-2 October, 2004, Sendai, Japan.
- [9] Comaniciu, D., Ramesh, V., & Meer, P. (2000) "Real-time tracking of non-rigid objects using mean shift", *Proceedings of IEEE Computer Vision and Pattern Recognition*, June 13-15, 2000, Hilton Head Island, South Carolina.
- [10] Nummiaro, K., Koller-Meier, E., & Van Gool, L. (2003) "An adaptive colour-based particle filter", *Image and Vision Computing*, Vol. 21, pp. 99-100.
- [11] Welch, G. & Bishop, G. (1997) *An introduction to the Kalman filter*, Department of Computer Science University of North Carolina at Chapel Hill.
- [12] Stigler, Stephen M. (1982) "Thomas Bayes's Bayesian Inference", *Journal of the Royal Statistical Society, Series A*, Vol. 145, pp. 250-258.
- [13] Sung, S. H. & Fan, C. H. (2005) *Detection and tracking multiple objects based on the particle filter technique*, National Taiwan Institute of Technology, Taipei.
- [14] Ho, Ming-Che, Chiang, Cheng-Chin, & Su, Ying-Yu (2012) "Object tracking by exploiting adaptive region-wise linear subspace representations and adaptive templates in an iterative particle filter", *Pattern Recognition Letters*, Vol. 33, pp. 500-512.
- [15] Bhattacharyya, A. (1943) "On a measure of divergence between two statistical populations defined by their probability distributions", *Bulletin of the Calcutta Mathematical Society*, Vol. 35, pp. 99-109.

Authors

Chao-Ju Chen received her B.A. degree from the Department of Oriental Languages at Tzu Chi University, Hualien, Taiwan in 2008. She received her M.S. degree from the Institute of Computer Science at National Hsinchu University of Education, Hsinchu, Taiwan in 2011. She is currently with National Tsinghua University, Hsinchu, Taiwan. Her research interests include image processing and computer visualization.



Wernhuar Tarnng graduated from the Department of Control Engineering at National Chiao Tung University, Hsinchu, Taiwan in 1984, and received his M.S. and Ph.D. degrees from the Department of Electrical and Computer Engineering, State University of New York at Buffalo, USA in 1987 and 1992, respectively. He is currently a professor at the Institute of e-Learning Technology at National Hsinchu University of Education, Hsinchu, Taiwan. His research interests include computer network, image processing, real-time systems, virtual reality, and e-Learning technology.



Kuo-Hua Lo received his Ph.D. degree from the Institute of Computer Science and Engineering at National Chiao Tung University, Hsinchu, Taiwan in 2013. He received his M.S. degree from the Department of Computer Science and Engineering at National Dong Hwa University, Hualien, Taiwan in 2006, and B.S. degree from the Department of Computer Science and Engineering, Tatung University, Taipei, Taiwan in 2004. He is currently with the Industrial Technology Research Institute, Hsinchu, Taiwan. His research interests include computer visualization and pattern recognition.

