Improved Particle Filter for Tracking Objects in Video

Tram Tran Nguyen Quynh[1]

***Abstract*—The method of tracking objects in computer vision has been studied for years, but until now it has been considered an open problem. However, nowadays,there is a method to track objects but its effectiveness has been proven in many studies around the world. It is recognized as a State-of-the-art, Particle Filter. In this paper, we improved some points in the Particle Filter to increase the accuracy of the algorithm. Then, we use the common datasets to build the experimental evaluations with the result tables to demonstrate the effectiveness of the improvements in algorithms.**

***Index Terms*— Particle Filter, Object Tracking.**

I. INTRODUCTION

Track the object through each frame of an image sequence is an important function in the computer vision applications. It is applied in the security field as camera tracking system such as monitoring and alerts to help supervisors for observing 24/7, detecting motion to warning of violations, detecting unusual situations based on motion recognition as affray, bank robbery, the risk of drowning. Besides, it also uses tracking vehicles on the roads such as early warning jams, recording the case gallop zigzagging, capturing cars to handle violations, etc.Another applications for tracking object for research and development with high usability is controlling auto-vehicle with camera system, GPS, sensors around the moving vehicleby perceptible computer, locate the lane, detect obstacles and other vehicles, identifying signage, thereby giving notice to the driver. In addition, some applications interaction between people and machines through movement.

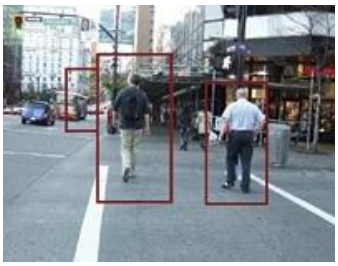

Fig. 1. Tracking pedestrians (source: IEEE Computer Vision and Pattern Recognition, 2007)

Currently there are many complex issues to be handled in the problem tracking objects. Therefore, many techniques have been developed to solve the problems of the problem, are classified into the following technical groups:image-based tracking, contour-based tracking, filtering-based tracking.

Image-based trackingextracts the common features and then groups them based on external information at a higher level. Typically, Intille et al. (1997) in [10] proposed a blob-tracker to track people in real time. Background was eliminated to obtain the foreground components. The area is then divided into foreground blobs based on color. This method is fast, but it has one major disadvantage of combining blobs when objects overlap each other.

Contour-based tracking uses the assumption that the object is determined by the road surrounded with some of the properties identified. Construction of the model shape (contour), dynamical model contour and other image parameters in the monitoring process. Yezzi and Soatto (2003)[5], Jackson et al. (2004) [6], and Rathi et al. (2005) [7]in their works proposed a definition for the distortion of the motion and shape to be able to apply for the distortion or movement of objects.

Filtering-based tracking has common methods such as Kalman Filter and Particle Filter. Kalman Filter uses the track shape and position over time in the linear system with Gaussian noise. These attributes really cause many obstacles in solving many problems in practice because the observed signal obtained quantities usually nonlinear and non-Gaussian distribution. Thus, Anderson and Moore (1979)[8] launched the Extended Kalman Filter algorithm (EKF). This algorithm is one of the best algorithms to solve nonlinear non-Gaussian. EKF algorithm works on the idea of linearization the observations obtained by the estimation of the item using Taylor expansions. However, in many cases, the estimated chain of EKF is modeling poorly in nonlinear functions and the interest probability distribution. It leadsto the algorithm will not converge. Julier and Uhlmann (1996)[9] propose an algorithm towards approximation a Gaussian-basedprobability distribution function, not any nonlinear distribution function. This algorithm was named unscented Kalman Filter (UKF). This algorithm has been shown to have better results EKF. However, the limit of UKF is that it cannot be applied in the problem with non-Gaussian distribution in general.

In contrast, Particle Filter can be applied in non-linear systems with Non-Gaussiannoises. The basic idea of Particle Filter is estimated probabilities in Bayesian theorem by a set of weighted samples.Typical research papers in Particle Filter are Isard in [3] with condensation algorithm, MacCormick and Isard (2000)[4] with Particle Filter with partitioned sampling, Rathi et al. (2005)[7] with Particle Filter with geometric active contours, etc.

In this paper, we will approach the problem of tracking object towards the integration of object recognition and object tracking with faceand pedestrian. We will conduct

Manuscript received December, 2015.

[1]***Tram Tran Nguyen Quynh***, *Foreign Languages and Information Technology Department, Ho Chi Minh City Vinatex Economic – Technical College, Ho Chi Minh City, Vietnam, Ho Chi Minh City, Viet Nam, +84909617505.*

ISSN: 2278 – 1323

detecting object, then tracking object. After a while, we will conduct object identifiers in order to increase efficiency tracking. We also improvethe number of tracking objects withthe same algorithm applying to the different Particle Filter group - each group corresponding to atracked object. Besides, we will improve the processing speed of the algorithm using parallel programming techniques and thread. In addition, we assessed the effectiveness of the improvements in the data obtained from renowned research paper to ensure the validity and correctness of the data. The data have feature richness and complexity: dataset "YouTube action" including videos that share some points as surroundings, with the angle of observation, etc. with the complexity of the dataset including major changes the camera moves, the objects appear and pose, scale objects, perspective, the complexity of the background scenery, lighting conditions, etc.Datasets "UCF Sports Action" includes video data at a resolution of 720x480 with the sports featured in a variety of different external.

The organization of this paper is as follows. In Section II and III, we introduce about object tracking and detection which are applied by the paper. Our proposed method is described in details in Section IV. Experiments results are discussed in Section V, followed by the conclusion and future work in the Section VI.

II. OBJECT TRACKING

A. *Object tracking problem*

The objective of tracking the object problem is to "understand" the motion of the object."Understanding" means information about objects such as position, velocity and other physical characteristics.

Most of the problem difficult to track the object is likely due to fluctuations of the video image. When an object moves through a viewport in the frame, the image of the object can change a lot. These changes comes from three main sources: the changing target positions (such as people standing posture switch; the car is going straight, turn left, etc.) or deformation of the target object, the change brightness, and the obscure part or all of the target (as when two people or vehicles passing by each other).

There are two main approachesof problem with pros and cons:(1) top-down approach which derived from the observer, perform extraction, segmentation of images or the input frame to find the object to be tracked such as Blob detection,Particle Filter; (2)bottom-up approach using theories of object and trying to test them using data obtained from the image, for example, template matching.

In this paper, we use Particle Filter method.

B. *The generalprocess of tracking objects*

Tracking objects can be divided into three main groups: (1) group objects have a common set of characteristics, such as cars, people, faces, etc. (2) group objects have a common set associated with a particular property such as moving cars, pedestrians, etc. (3) group objects share the same specific attributes such as the moving object, any objects that users selected in the first frame.

Object tracking algorithm actually finds an image area moving from start frame to other frames associated with objecthaving its own features. In general, we have the following main steps. First, we need to build a reference model to describe the object to be tracked. Then on each input frame, based on similarity measure, the algorithm searcheslocalize area which is the most similar to the reference model.

Reference model is the model describes the information about the "appearance" of the object to be tracked. There are many ways to build reference models for objects as gray-level models, contour models, and most commonly used in applications that are used to track the object is color model. However,there some of the issues raised.

We should use which color system such as RGB, HSS, etc. Note that when we use color models to reference model means that we have to add one suppose that we just keep track of objects on color image rather is any image. In addition, it should be carefully chosen color system because it is very sensitive to light, the scenery. Currently in practical applications are using Hue-Saturation color system-Value (HSV) and how to use the distribution model. There are many ways to create distribution model as Gaussian distribution, or Gaussian Mixture, or simply as a histogram. Practical applications are using the histogram distribution model.

To compare the candidate model and the reference model in each input frame, we need to have a function to calculate the similarity measure. This function is responsible to calculate the degree of similarity between two objects on which then determines the state of the object to be tracked. For example, the function SSD (Sum of Squared Differences) is used in case the condition that constant brightness light mean value of the pixels do not change from frame to other frames and function SAD (Sum of Absolute Differences).

C. *Particle Filter*

Object tracking using Particle Filter is based on a probabilistic approach, using the equations to predict the state of the object and the equation to update based on the knowledge gathered from the observations on the object to correct the previous estimate of the object's status.

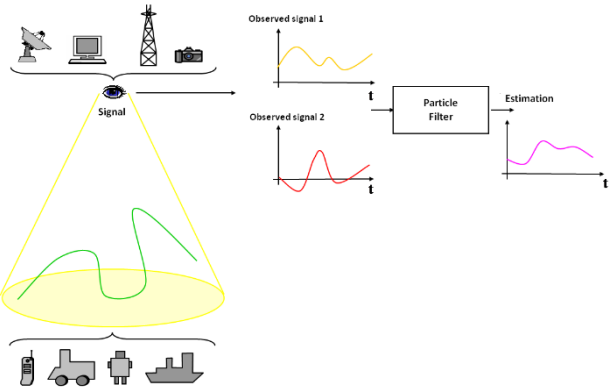

Fig. 2. The process from the actual signal to the estimation

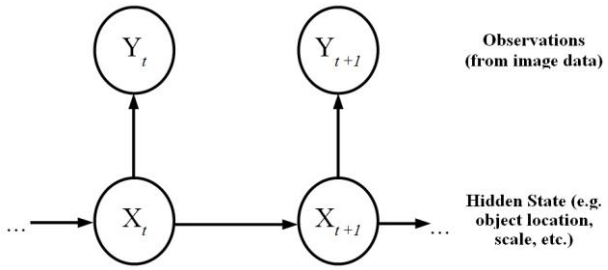

Fig. 3. Particle Filter probability model

ISSN: 2278 – 1323

This method uses theory of the estimate Bayesian regression, and the idea of Monte Carlo method combined with sampling factors to approximate the solution. The implementation of this approach is to build dynamic model and model observation with applying probabilistic models based reference model of the object to analyze the data stream from video.Particle Filter calculates approximation of posterior density function.But unlike other methods based on the analysis math, trying to find a solution of the equations on through one or more other equations, the Particle Filter uses a large set of sample data generated from the distribution functions.

1) Definition

Object tracking technique is based on the previous state of the object to predict object locations in the next frame. The following terms are defined:

- x_t: The state of the object at time t.
- z_t: The signal from the data observed from the current frame in the video sequence at time t.
- $(\mathrm{x}_1, \mathrm{x}_2 \ldots \mathrm{x}_t)$: The set of the object state from the start to the time t.
- $(\mathrm{z}_1, \mathrm{z}_2 \ldots \mathrm{z}_t)$: The set of data observed to time t.

2) Dynamics Model

The dynamic model of the object is the probability equations describing motion, the state changes of object in the system. In practical applications, the state of the object is described by a vector $\mathrm{x}_t = (\mathrm{x}, \mathrm{y}, \mathrm{s})$ where x, y are the coordinates of the rectangle containing the object, s is the scale of the object.

Spreading the sample collection and estimating the movement of objects uses a second order autoregressive dynamic model as follows formula:

$$\begin{cases} x_{t+1} = Ax_t + Bx_{t-1} + Cv_t, v_t \sim N(0, \sigma_t) \\ y_{t+1} = Ay_t + By_{t-1} + Cv_t, v_t \sim N(0, \sigma_t) \\ s_{t+1} = As_t + Bs_{t-1} + Cv_t, v_t \sim N(0, \sigma_t) \end{cases} \quad (1)$$

Where matrix A and B represent invariant components and matrix C represents the random component. These ingredients may be selected from the sample set can be assigned school or manually. Probability σ_t is Gaussian noise.

3) Observation model

Observation model is the basis for the measurement and calculating probability for the probability equations of system. The purpose of observation step is to measure the ability of each sample to predict likelihood with the observed data, which characterized through the weight of sample. The model used to observe the features of the image, such as edges, color, chart (histogram), etc., to determine the estimationstate based on the sample input data or observation data. In practical applications, we use observation models based on color model in the color system HSV.

The color model is obtained by histogram technique in the HSV color system to extract color information from the shadow effect. However, the color information is only reliable when two values Saturation and Value are not too small. Therefore, we build a histogram HS with bins N_h, N_s by using only the pixels with two values Saturation and Value greater than two levels of 0.1 and 0.2 respectively. However, the remaining pixels "colorless" could be very important information when tracking the colored areas are mostly black and white. So, the obtained histogram includes the number of bins $\mathrm{N} = \mathrm{N}_h \mathrm{x}\, \mathrm{N}_S + \mathrm{N}_v$.

At time t, the histogram model $h(x_t)$ of the state vector x_t will compare with the reference model $h_0 = \{h_0(n)\}_{n=1\ldots N}$. Reference model built at the beginning time t0 can help user to select the tracking object or combined with the detection module to classify object automatically.

$$h_0 = h(x_{t_0}) \quad (2)$$

Calculating the likelihood quantity, we need to identify the level of "proximity" between the candidate histogram model and reference models through distance D on HSV color distribution. Distance D is calculated by Bhattacharyya distance as follows:

$$D[h_0, h(x_t)] = 1 - \sum_{n=1}^{N} \sqrt{h_0(n) h(n; x_t)} \quad (3)$$

To calculating weight, we use the below formula:

$$w_t = p(z_t | x_t) \propto e^{-\lambda D^2 [h_0, h(x_t)]} \quad (4)$$

Based on the empirical evaluation, λ is considered the best value of 20. For the number of histogram bins, we use default settings $N_h = N_v = N_s = 10$ (10 bin for each color system Hue, Saturation and Value). So, there are the total of bins N = 10x10 + 10 = 110 bins.

4) Resampling algorithm

From the theory of resampling method, we have re-sampling algorithm as the next step:

- In the first frame, create a set of N samples with weighting is 0.

  In the next frame k:

- Normalizeweighted value using formula:

$$w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_{j=1}^{N} \tilde{w}_t^{(j)}};\ 1 \le i \le N \quad (5)$$

- Calculate the estimated sample according to the following formula:

$$\tilde{N}_{eff} = \frac{1}{\sum_{i=1}^{N} (w_t^{(i)})^2} \quad (6)$$

- If $\tilde{N}_{eff} < N_{thr}$, create a new set from N current samples with weighted value:

$$w_k^{(i)} = \frac{1}{N}, 1 \le i \le N \quad (7)$$

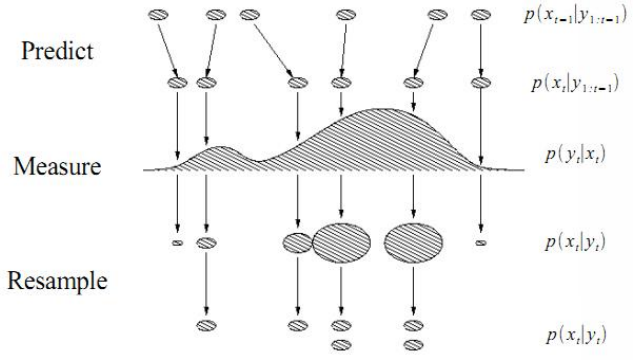

Fig. 4. The three main steps in the algorithm Particle Filter

The steps in the process of determining the status of the object through the Particle Filter:

- Initialize the object state x_0 from the first frame.
- Create a sample of N samples $\{s_t^i\}_{i=1,2,\ldots,N}$.
- Predict x, y, s for a sample using dynamic models.
- Calculate the weight of a sample by observation model.

ISSN: 2278 – 1323

- Resampling
- Determining the status of the object at the present time based on the average of N / 2 particleswith the largest weighted value.

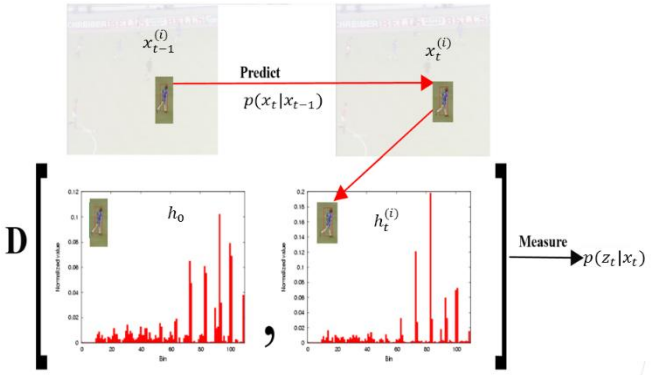

Fig. 5. Calculation steps on a video.

There are three important steps:

- Prediction: predict the state of the object at time t based on dynamic model.
- Measure: calculate the sample weights in the set based on observations (signals from video - compare the histogram of the sample) at the current time t to choose a sample "similarity" with the object.
- Resample: resampling to avoid sample degradation from the current set of sample files to create a new form with not too small weights.

III. OBJECT DETECTION

In this section, we'll studied about object detection applied in practical applications on two subjects that face and pedestrians.

A. Object detection problem

There are many methods of detecting objects.Based on the properties of the method, we can be divided into four main approaches as follows:(1) knowledge-based method encoding human understanding of the types of objects and create sets of rules to determine the object, (2) the approach based on the features do not change with the goal to find out the features described object structures (features unchanged posture, placement of television devices, brightness change, etc.), (3) matching-based method using use the standard form of the object, then performing matches on input data, (4) pattern-based method using training from a set pattern to identify the object.

In this paper, we uses AdaBoost (pattern-based approach) applying the strong classifier with combination of weak classifier based on the featured Haar-like to identify the object.

B. Haar-like feature

According to Viola and Jones in [12],there are four basic characteristics to identify the object. Each Haar-like feature is a combination of two or three rectangular "white" or "black" asFig. 6.

Fig. 6. Four basic Haar-like features

Using these features to identify human faces, fourbasicHaar-like featured is expanded and divided into three feature sets:edge features, line features, and center-surround features.

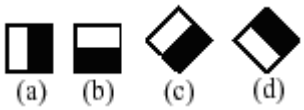

Fig. 7. Edge features

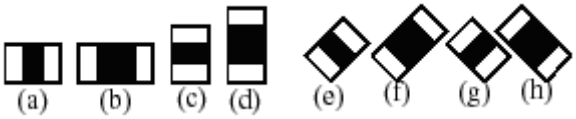

Fig. 8. Line features

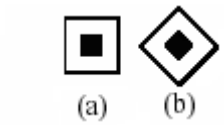

Fig. 9. Center-surround features

Using the features above, we can calculate the value of Haar-like feature which is the difference between the sum of the pixels of the black areas and white areas:

$$f(x) = \sum Blackpixels - \sum Whitepixels \quad (8)$$

Using this value, compared with the crude pixel value, Haar-like features can increase or decrease changes in-class or out-of-class object for helping classification easier.

C. Integral image

Calculating the value of Haar-like features, we must calculate the sum of the pixels in the image area. But to calculate the value of the Haar-like feature for all positions on the image, it requires high computing costs, does not meet the real-time applications. Therefore, Viola and Jones launched a concept called Integral Image, which is a two-dimensional array with a size equal to the size of the image to calculate the Haar-like features with each element of the array calculated by aggregating upper pixel (row-1) and left (column-1) of it. Starting the upper left to the lower right position of image, this calculation uses the integer addition, so the execution speed is very fast.

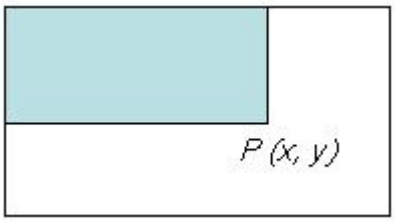

Fig. 10. The value at point (x, y) in Integral image with the sum of the pixels in the upper left

Equation of calculating Integral image:

$$P(x,y) = \sum_{x' \le x, y' \le y} I(x', y') \quad (9)$$

Where P(x, y) is Integral image, I (x', y') is original image.

Supposing need to calculate the total of the gray level in region D as inFig. 11, we can use the equation below:

$$D = A + B + C + D - (A + B) - (A + C) + A \quad (10)$$

Where A + B + C + D is the value at the point P_4 on Integral Image, likewise A + B is the value at the point P_2, A + C is the value at the point P_3, and A is the value at the point P_1.

ISSN: 2278 – 1323

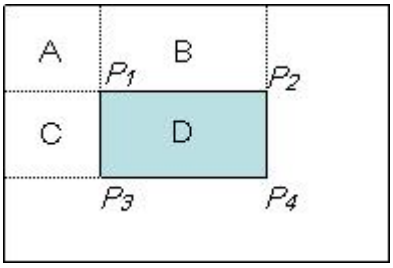

Fig. 11. Calculating quickly sum of the grayscale values of the region D

We can rewrite the above equation to calculate D as follows:

$$D = \underbrace{(x4, y4)}_{A+B+C+D} - \underbrace{(x2, y2)}_{(A+B)} - \underbrace{(x3, y3)}_{(A+C)} + \underbrace{(x1, y1)}_{A} \quad (11)$$

D. AdaBoost

Next, to choose the Haar-like feature for setting threshold, Viola and Jones uses a machine learning method called AdaBoost. AdaBoost will incorporate the weak classifiers to form a strong classifier.

AdaBoost is a strong complex nonlinear classifier based boosting approach launched by Freund and Schapire in 1995 [13].AdaBoost uses weight to mark the pattern recognizing hardly. During training, each weak classifier is built. The algorithm will proceed to update the weights to prepare for the building of the next weak classifier: increase the weight of the sample misidentified and reducing the weight of the samples identified properly by the weak classifier just built. By this way, the next classifier could focus on the previous classifier which did not do well. Finally, the weak classifier will be combined depending on the well severity to make the strong classifier, as equation below:

$$H(x) = sign\big(a_1 h_1(x) + a_2 h_2(x) + \cdots + a_n h_n(x)\big) \quad (12)$$

Where $a_t \geq 0$: standardized coefficient for weak classifiers.

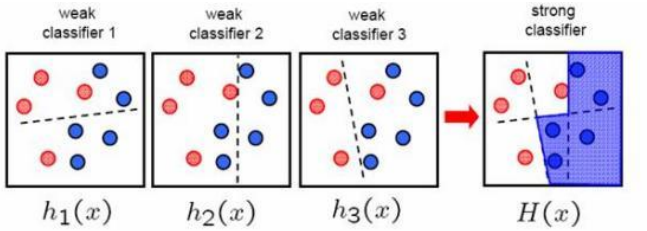

Fig. 12. Combine the weak classifiers into strong classifier.

E. Process of object detection

From the original image, we create Integral Image to calculate quickly the sum of the values of a grayscale on any rectangular area of the original image. The sub-image area will be taken through the basic Haar functions to estimatefeatures. The estimated results will be taken through the AdaBoost to eliminate quickly features which it is not features of the object. Only a subset of the features that the AdaBoost regulator detects it is capable to be human face features, will transfer to the resultregulator (a set of weak classifier). The regulator confirmsit is the object to determine if the results of the weak classifier confirmed that it is the object to be identified.

Each weak classifier will determine the outcome of a Haar-like feature with threshold low enough due to overcome all of the sample data in the training data set. In the process of identifying objects, each of the sub-imageswill be checked features of the sequence of Haar-like feature. If there is a Haar-like feature available to confirm that the object to determine, other features need not to process. Ordering at the features in a series of Haar-like features will be based on the weight of the feature which AdaBoost decides based on the number and order of appearance of the Haar-like features.

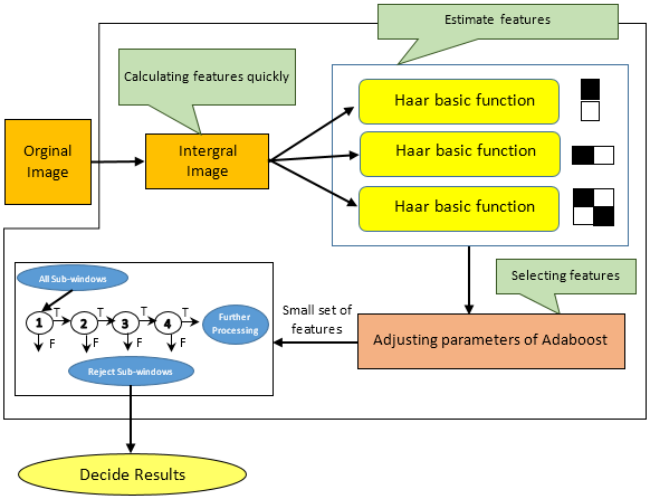

Fig. 13. The process of detecting objects

IV. PROPOSED METHOD

We build an application based on open source of Professor Rob Hess, Oregon State University - USA, 2002 in[11],[14] for an application for tracking objects with the following characteristics: applying Particle Filter algorithm to track objects in video, using the OpenCVlibrary for image processing. While it is a new breakthrough, but there are some limitations such as:calculation and display the result on the screen in the same process, making the speed is reduced significantly; only one object tracking by user-specified.

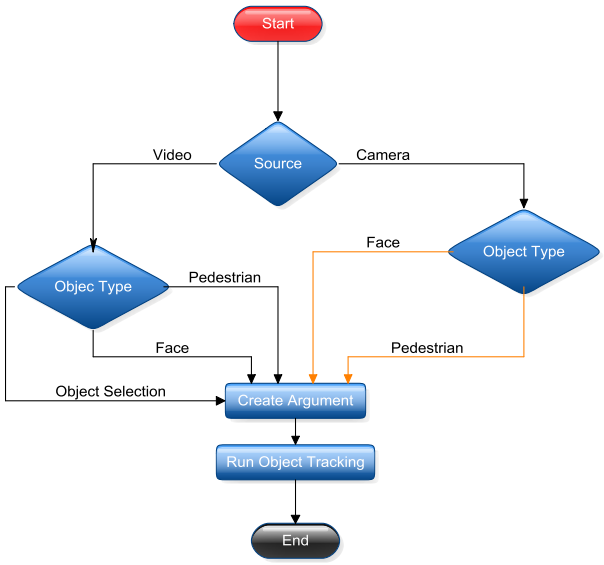

Fig. 14. Flowchart of main process

In this paper, the goal is to create a useful application and practical reference source such as applying Particle Filter algorithm to track objects in video and camera, tracking the object and save traces its path, identification pedestrians and human facecombined with tracking objects by camera. Fig. 14 shows main process in this paper including detecting object such as face or pedestrian. Face recognition is relatively accurate. But the identification of pedestrians back to results very low: only identify pedestrians to look in a direction perpendicular to the surface and back.

We have built applications with some results as follows:

ISSN: 2278 – 1323

enhancing to track multiple objects specified by the user, face recognition and pedestrian recognition combined with tracking multiple objects, applying parallel programming to take advantage of processor.

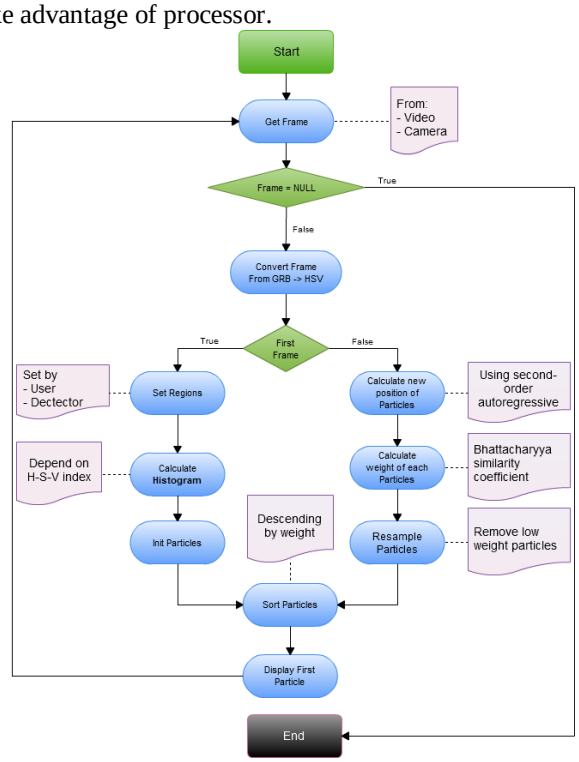

Fig. 15. Flowchart for tracking objects using Particle Filter algorithm

V. EXPERIMENTS AND DISCUSSION

This section provides the basis for the data and formulas to analyze and evaluate the effectiveness of the algorithm to detect and track objects. First we need to know two following definition of ground truth information is tracked container object in each frame. Similarly, the results of the algorithm are to track container object in each frame.

Technical evaluation in [2] determined limit from distance matrix between the focus of targeting framework between the ground truths and the tracking results of the algorithm. This limit is used to find the correspondence between the results of the algorithm to track and ground truth with calculating the quantities such as False Positive Track Error, False Negative Track Error, Average Area Error, and Task Incompleteness Factor. However, these quantities do not measure the performance of the tracking algorithm in case there is an overlap between the objects.

The performance evaluation data in [1] is divided into the data based on the frame and the object. For data based on frame: true positive, true negative, false positive, and false negatives are calculated for all the frames, and is used to calculate Tracker Detection Rate, False Alarm Rate, Detection Rate, Specificity, Accuracy, Positive Prediction, Prediction Negative, False Negative Rate, and False Positive Rate. For object-based data, each separate object is used to calculate true positive, false positive and total ground truth to calculate Tracker Rate Detection, False Alarm Rate, and Object Tracking Error (OTE). Quantities based on the frame to provide information about how the algorithm handles tracking objects in a frame, and object-based data for performance measurement tracking algorithm on each object in time of video. Within the scope of this paper, we are applying based frame data in the performance evaluation of tracking algorithm.

To perform our evaluation should determine the degree of overlap between the ground truth and the result of the algorithm. It is simplest to consider if the focus of one of the two frames in the rest frame; from which the following values are determined:

- TN (True Negative): the number of frames on the application and ground truth the object does not appear.
- TP (True Positive): the number of frame and ground truth on application objects appear and object container coincidental.
- FN (False Negative): the number of frames on the ground truth object appears, while tracking the wrong application object.
- FP (False Positive): the number of frames on the application objects appear but ground truth is not.
- TG (Total ground truth): the total number of frames which objects appear on a ground truth.
- TF (Total frame): the total frames of a video sequence.

$$\text{Tracker Detection Rate (TRDR)} = \frac{\text{TG}}{\text{FP}} \quad (13)$$

$$\text{False Alarm Rate} = \frac{\text{FP}}{\text{TP}+\text{FP}} \quad (14)$$

$$\text{Detection Rate} = \frac{\text{TP}}{\text{TP}+\text{FN}} \quad (15)$$

$$\text{Specificity} = \frac{\text{TN}}{\text{FP}+\text{TN}} \quad (16)$$

$$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{TF}} \quad (17)$$

$$\text{Positive Prediction} = \frac{\text{TP}}{\text{TP}+\text{FP}} \quad (18)$$

$$\text{Negative Prediction} = \frac{\text{TN}}{\text{FN}+\text{TN}} \quad (19)$$

$$\text{False Negative Rate} = \frac{\text{FN}}{\text{FN}+\text{TP}} \quad (20)$$

$$\text{False Positive Rate} = \frac{\text{FP}}{\text{FP}+\text{TN}} \quad (21)$$

In above values, there are three values Specificity, Accuracy, Positive Prediction which are indicators of interest of all. In this paper, we only use Accuracy to present the results of each data.

The evaluation of a tracking system based on measuring the performance of tracking algorithms related to the features of the system. The basic principle of a tracking system is to detect objects and exploits orbital objects in a video footage. For a tracking algorithm to monitor, exporting orbitsof multiple objects, tracking algorithms can differentiate between objects.

The criteria for assessing how well a tracking algorithm:detecting all movement in and out of sight, distinguishing between objects displayed in a scene at a time, tracking and extracting the orbits of all assigned objects with being maintained for all objects tracked, motion or non-motion of the objects not to lose track of the object, handling the overlap between subjects and exposure while keeping track of the object.

A. *Experimental Database*

We use three dataset as follows:

- YouTube action dataset[15]: including 11 categories: throwing basketball, cycling, diving, horse riding,

ISSN: 2278 – 1323

juggling balls, swinging, tennis, jumping trampoline, volleyball, and walking with the dog, etc. For each category, the videos are grouped into 25. The video clip of the same groups that share some points as surroundings, with view to observe, etc. The complexity of the database: a major change in motion camera, the object appears and sets, object size, perspective, background clutter, lighting conditions, etc.

- UCF Sports Action Dataset[16]: This dataset includes a set of activities gathered from various sports on TV channels such as BBC and ESPN. The database consists of video at 720x480 resolution, describes the typical activities apply in a variety of different outdoor scenes.
- Boston Head Tracking Database[17]: This dataset is done in the lab, each video has duration of about 6 seconds and record the face with all the state includes: moving, glasses looked up and down and to the environmental light changes.

B. Results

TABLE 1. RESULTS OF TRACKING OBJECTSWITH SELECTING BY USER

#	Video	Frames	Accuracy	
			This Paper	Robert Hess
1	v_biking_01_02	151	100	97.35
2	v_biking_01_03	151	88.08	93.38
3	v_biking_03_03	179	91.62	88.83
4	v_biking_03_04	191	91.62	74.35
5	v_biking_06_05	169	87.57	80.47
6	v_biking_12_04	201	100	97.51
7	v_biking_16_04	151	80.13	99.34
8	v_jumping_01_03	201	88.06	87.06
9	v_jumping_01_04	201	94.03	98.51
10	v_jumping_02_03	201	48.76	48.26
11	v_jumping_02_04	201	51.74	63.18
12	v_jumping_03_01	201	84.08	67.16
13	v_jumping_03_04	201	100	100
14	RunSide001	64	100	100
15	RunSide002	64	100	100
16	RunSide005	64	100	100
17	RunSide006	64	100	100
18	RunSide007	64	100	100
19	SkateBoard001	69	100	100
20	SkateBoard002	69	100	100
21	SkateBoard004	69	100	100
22	SkateBoard011	69	100	100
23	v_spiking_07_02	87	75.86	3.45
24	v_spiking_07_04	67	100	16.42
25	v_spiking_07_06	57	42.11	19.3
26	v_spiking_07_07	67	79.1	86.57
27	v_shooting_13_02	101	94.06	88.83
28	v_shooting_13_04	111	54.95	74.35
29	v_shooting_14_05	75	97.33	97.33
30	v_shooting_16_01	141	33.33	32.62
31	v_shooting_16_02	103	91.35	78.85
32	v_shooting_19_07	120	90.83	90.0
33	v_walk_dog_05_03	201	100	18.41
34	v_walk_dog_05_04	121	100	100
35	v_walk_dog_05_05	113	100	100
36	v_walk_dog_09_01	173	98.27	100
37	v_walk_dog_09_02	201	95.52	100
38	v_walk_dog_10_03	201	100	100
39	v_walk_dog_10_04	201	100	100
40	v_walk_dog_10_05	201	100	100
41	v_walk_dog_13_02	201	100	97.01
42	v_walk_dog_14_03	181	100	93.92
Total		**5718**	**89.49**	**75.25**

TABLE 2. RESULTS OF TRACKING HUMAN FACE

#	Video	Frames	Accuracy
1	jam1	199	83.42
2	jam2	199	100
3	jam4	199	100
4	jam5	199	100
5	jim1	199	100
6	jim2	199	95.48
7	ssm1	199	100
Total		**1393**	**96.99**

TABLE 3. RESULTS OF TRACKING PEDESTRIANS

#	Video	Frames	Accuracy
1	v_walk_dog_01_04	151	74.83
2	walk002	100	56
3	walk003	100	69
4	walk008	101	11.88
5	walk010	102	76.47
6	walk014	100	100
Total		**654**	**64.7**

VI. CONCLUSION AND FUTURE WORK

In this paper, we has overcome limitations such as monitoring speed by splitting parallel processing; track multiple objects simultaneously combined with detection and tracking human face, pedestrians.The monitoring objects and faces relatively accurate. But the paper also has some limitations:the monitoring of large-sized object will cause slower processing speed by calculating the histogram, detect and track pedestrians without high accuracy due to the detection process.

The problem of tracking objects also has many open and unresolved issues, as tracked on multiple cameras; on a variety of video (the video of the news; video contains more noise, no structured format, contains many views, ...). So, it will continue to be developed in the coming years.

REFERENCES

[1] A. Yilmaz, O. Javed and Mubarak Shah, "Object Tracking: A Survey", *ACM Computing Surveys*, 2006.
[2] D. Koller, J. Weber and J. Malik, "Robust Multiple Car Tracking with Occlusion Reasoning", *California Partners for Advanced Transit and Highways*, Institute of Transportation Studies, UC Berkeley, 1994.
[3] M.Isard and A.Blake, "Condensation – conditional density propagation for visual tracking", *International journal of computer vision*, 1998.
[4] J.MacCormick, M.Isard, "Partitioned sampling, articulated objects, and interfacequality hand tracking", *In: ECCV. pp. 3–19*, 2000.
[5] A.Yezzi, S.Soatto, "Deformotion: Deforming Motion, Shape Average and the Joint Registration and Approximation of Structures in Images", *Int'l J. Computer Vision*, 2003, vol. 53(no. 2):153–167.
[6] J.Jackson, A.Yezzi, S.Soatto, "Tracking Deformable Moving Objects under Severe Occlusions", *Proc. IEEE Conf. Decision and Control*, 2004.
[7] Y.Rathi, N.Vaswani, A.Tannenbaum, A.Yezzi, "Particle Filtering for Geometric Active Contours with Application to Tracking Moving and Deforming Objects", *Proc. Conf. Computer Vision and Pattern Recognition*, 2005.

ISSN: 2278 – 1323

[8] B.D.O.Anderson, and J.B.Moore, "Optimal Filtering", *Englewood Cliffs*, New Jersey, Prentice–Hall, 1979.
[9] S.J.Julier and J.K.Uhlmann, "A General Method for Approximating Nonlinear Transformations of Probability Distributions", *Technical report, Department of Engineering Science, University of Oxford, OX1 3PJ UK*, 1996.
[10] S.S.Intille, J.W.Davis, and A.F.Bobick, "Real-Time Closed-World Tracking", *Conference on Computer Vision and Pattern Recognition*, 1997.
[11] Rob Hess, "Discriminatively Trained Particle Filters for Complex Multi-Object Tracking", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, CVPR Workshops 2009.
[12] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001.
[13] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting",*Journal of Computer and System Sciences*, 55(1):119–139, August 1997.
[14] http://blogs.oregonstate.edu/hess/
[15] http://crcv.ucf.edu/data/UCF_YouTube_Action.php
[16] http://crcv.ucf.edu/data/UCF_Sports_Action.php
[17] https://www.cs.bu.edu/groups/ivc/HeadTracking/Home.html

Tram Tran Nguyen Quynhreceived the Bachelor degree in information technology from Ho Chi Minh City University of Foreign Language – Information Technology, Ho Chi Minh City, Vietnam, in 2005.

In 2010, she has joined the Faculty of Foreign Languages and Information Technology, Ho Chi Minh City Vinatex Economic – Technical College, Ho Chi Minh City, Vietnam, where she is currently a Lecturer. Her research interests include computer vision, image processing, and parallel programming.

ISSN: 2278 – 1323