

## **An Efficient Object Tracking Algorithm based on Dynamic Particle Filter**

**V. Raudonis, R. Simutis**

*Department of Process Control, Kaunas University of Technology, Lithuania  
Studentu str. 50, Kaunas, phone: 300296, e-mail: rimvydas.simutis@ktu.lt*

**A. Paulauskaitė–Tarasevičienė**

*Department of Business Informatics, Kaunas University of Technology, Lithuania  
Studentu str. 56, Kaunas, phone: 300378, e-mail: agne@ifko.ktu.lt*

### **Introduction**

The tracking of moving non-rigid objects in the video sequence is frequently solved task in the computer vision systems. For this purpose dynamic models are used which describes the motions of the object or the algorithms which measures certain features of the objects, i.e., roundness, color or edge. The measure of the features or motions is not impossible without the impact of the noise. Therefore, every measure can be valued as probability measure. When the motion or the feature of the object can be described with linear model, then for the tracking purposes linear methods can be used [1-3]. However, many object tracking cases are non-linear, which can be solved using a particle filter.

The particle filter is the algorithm based on the probability measure, which do not use the probability density function directly, but approximates it by certain number of particles. Each particle is described with the appearance position and its weight, which is the certain measure of probability or similarity. Such approximation of the probability density function is called a sequential Monte Carlo method [4], as well. The particle filter algorithm is recursive in nature and comes through three stages: evolution, re-sampling and propagation. After every computation each particle is newly modified and propagated based on the latest measures and the motion model. During the re-sampling process, the particles with smallest weight values are eliminated.

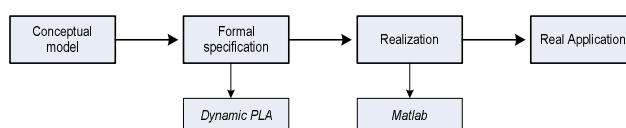
The particle filter can be used in the various object tracking systems, i.e., car detection and tracking, vision based security systems etc [6]. In the work the adaptive particle filter was proposed which was used for the accidently appearing face recognition [5]. Authors of this paper have used two time series models and one presumption for the face recognition and tracking purposes. The first adaptive model describes the process of the motion and appearance from the video data, second model describes the changes between tracked face and the

database of the faces. The presumption is taken in the account that all faces in the database are in the frontal view.

The color features of the observable object are frequently used for the description of the tracking target and the separation of the foreground from the complex background. Color-based tracking can be implemented using color density images [7], color histograms [8] and etc [10-12, 16-18]. The tracking algorithm proposed in this paper is similar to the CONDENSATION algorithm [9] which is dedicated to track the object contour. However, the algorithm presented in this paper is able to track the color features of the observable object. The Gaussian random noise is used as motion model for the object movement prediction. The particles distributed according to such motion model can follow any movement of the object.

In this paper, the tracking problem of the non-rigid objects in the complex background is solved using dynamic particle filter, where the number of particles relays on the changing rate of the object location. The proposed approach automatically uses the less number of particles when the position of the object does not change rapidly from one frame to another frame and more particles when the object moves relatively fast.

Since, the algorithm of the object tracking is quite complicated, there is a need for a formal description and validation of the algorithm. The dynamic PLA notation – DPLA [15] was chosen for the formal specification of the proposed algorithm, which has a possibility to define the variation of the number of particles in the solving tasks.



**Fig. 1.** The functional scheme of the design of the dynamic particle filter

The paper is organized as follows. Second section explains the proposed dynamic particle filter in the conceptual and formal manners. In the third section introduces experimental setup and results. Finally, conclusions and future guides are presented in the last section.

### Conceptual model of the dynamic particle filter

This section explains briefly the proposed approach of dynamic particle filter. As it was mentioned, the particle filter comes through three states: evolution, re-sampling and propagation. Each state is executed in sequence. The weights of each generated particle are computed based on the measure model during evolution state. The measure model uses the correlation between template and selected HSV color histograms in this paper (see Fig. 2). If histogram of the particle is similar to the template histogram then the weight of this particle is set to 1, and otherwise to 0. The particles with smallest weight values are eliminated during the re-sampling state. New particles are generated according the Gaussian random noise and the information about the last object location during propagation state. The real location of the object is the stable point of the particles cloud. The object location which was found in the frame  $t - 1$  is used as central point for particle propagation.

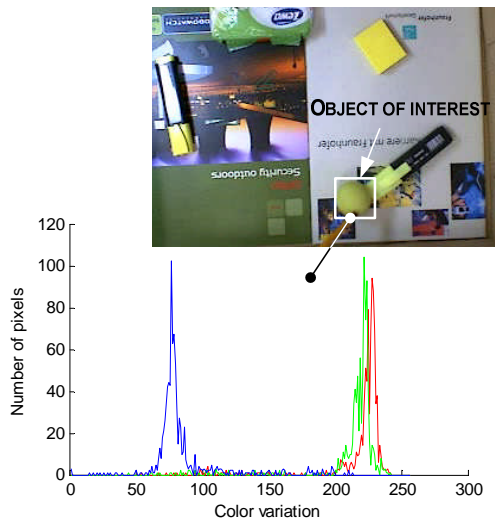


Fig. 2. The color histogram of the object of interest

### Formal description of the dynamic particle filter

The algorithm of dynamic particle filter was formalized using formal DPLA (dynamic PLA) notation [15], which is extension of PLA [14]. Dynamic PLA is able to various dynamic changes of systems. Proposed algorithm in dynamic PLA specification was described using internal events of decision making (DM) aggregate, which decompose dynamic particle filter into the seven stages:

- $e_1''$  – The number selection of particles;
- $e_2''$  – The estimation histograms and positions of generated particles;
- $e_3''$  – The comparison of histograms;

- $e_4''$  – The selection of proper particles;
- $e_5''$  – The estimation of max distance among selected particles;
- $e_6''$  – The selection of proper group of particles using clustering;
- $e_7''$  – The estimation of the new average coordinates.

The particles  $d_i$  in dynamic PLA specification are interpreted as the elements of the set  $D(t_m)$  which is defined in discrete component of aggregate  $DM$  (see Fig. 3). During the particular internal events to the particles the new attributes as coordinates, histograms and correlation coefficients are added.

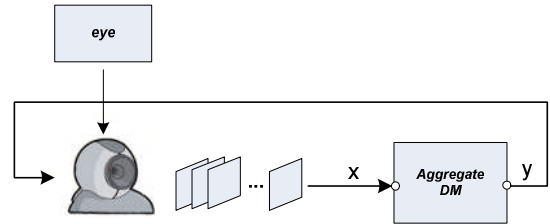


Fig. 3. DPLA model of dynamic particle filter

The initial state  $Z(t_0)$  of the aggregate  $DM$  is described below, which includes all attributes of the aggregate in initial time moment:

$$Z(t_0) = (X(t_0), Y(t_0), E'(t_0), E''(t_0), v(t_0), z_v(t_0), G(t_0), H(t_0), A(t_0), R(t_0)), \quad (1)$$

where  $X(t_0) = \{x\}$  – the set of input signals,  $Y(t_0) = \{y\}$  – the set of output signals,  $E'(t_0) = \{e'_i\}$  – the set of external events,  $E''(t_0) = \{e''_i\}, i = \overline{1,7}$  – the set of internal events,  $z_v(t_0) = \{w(e''_i, t_m)\}, i = \overline{1,7}$  – continuous component.

Discrete component:

$$v(t_0) = \{D(t_0), k(t_0), \sigma_x(t_0), \sigma_y(t_0), C(t_0), MAX(t_0), reg, red, exp, \beta, \alpha, \mu_r\}, \quad (2)$$

where  $D$  – the set of particles;  $k$  – the number of picture;  $C$  – the class of particles at initial time;  $MAX$  – maximum distance between particles;  $\sigma_x, \sigma_y$  – standard deviations;  $\alpha = const$  – threshold value of the distance;  $reg = const$  – regular number of particles;  $red = const$  – reduced number of particles;  $exp = const$  – expanded number of particles;  $\beta = const$  – threshold value of correlation coefficient;  $\mu_r = const$  – threshold value of movement.

The set of transition operators:

$$H(t_0) = \{H(e'_i), H(e''_i)\}, i = \overline{1,7}. \quad (3)$$

The set of output operators:

$$G(t_0) = \{G(e'_i), G(e''_i)\}, i = \overline{1,7}, \quad (4)$$

where  $A(t_0) = \{\emptyset\}$  – the set of initial aggregates,  $R(t_0) = \{\emptyset\}$  – the set of internal links.

As external event  $e_1''$  aggregate  $DM$  gets the picture (input signal  $x$ ) generated from the camera. Whenever processing of the picture is finished aggregate sends the output signal  $y$  back to the camera and asks for another picture. With the first input  $k(t_m)=0$  signal the average coordinates  $\mu_x, \mu_y$  of initial position and the template histogram  $p^{(j)}$  of tracking object are determined and added to the discrete component of aggregate. Finally the first internal event  $e_1''$  is invoked despite the value of variable  $k$ . The transition operator of external event is defined below:

$H(e_1'')$ :

$$w(e_1'', t_m + 1) = t_m + \eta_k^1, \quad k(t_m) \geq 1. \quad (5)$$

$$\begin{cases} v(t_m + 1) = v(t_m) \cup p^{(j)} \cup \mu_x(t_m) \cup \mu_y(t_m), \\ w(e_1'', t_m + 1) = t_m + \eta_k^1, \\ k(t_m + 1) = k(t_m) + 1. \end{cases} \quad \text{if } k(t_m) = 0 \quad (6)$$

The description of the process of the tracking object is presented by transition operators of internal events, performed by appropriate order.

The selection of the number of the particles - primarily performed internal event  $H(e_1'')$ :

$$\begin{cases} w(e_2'', t_m + 1) = t_m + \eta_k^2, \\ D(t_m + 1) = \{d_i\}, \quad i = \overline{1, n(t_m)}, d_i(t_m) = \{\emptyset\}, \end{cases} \quad (7)$$

where  $D(t_m)$  - the set of particles.

The number  $n$  of particles can be regular *reg*, reduced *red* or expanded *exp*. It is chosen in accordance with position shift of tracking object from two contiguous results. The position shift

$$ps = |\mu_x(t_m - 1) - \mu_x(t_m)| + |\mu_y(t_m - 1) - \mu_y(t_m)| \quad (8)$$

is compared with  $\mu_r$ , which value depends on the ration of the observable object and total image. This value was determined experimentally in our example.

$$n(t_m) = \begin{cases} reg, & \text{if } ps > \mu_r, \\ red, & \text{if } ps \leq \mu_r, \\ exp, & \text{if } \mu_x(t_m) = \mu_y(t_m) = \infty. \end{cases} \quad (9)$$

Compared with this value standard deviations  $\sigma_x(t_m)$  and  $\sigma_y(t_m)$  are selected as well.

$$\begin{cases} \sigma_y(t_m) = \sigma_{y_{reg}} \wedge \sigma_x(t_m) = \sigma_{x_{reg}}, & \text{if } ps > \mu_r(t_m), \\ \sigma_y(t_m) = \sigma_{y_{red}} \wedge \sigma_x(t_m) = \sigma_{x_{red}}, & \text{if } ps \leq \mu_r(t_m), \\ \sigma_y(t_m) = \sigma_{y_{exp}} \wedge \sigma_x(t_m) = \sigma_{x_{exp}}, & \text{if } \mu_x(t_m) = \mu_y(t_m) = \infty, \\ w(e_2'', t_m + 1) = t_m + \eta_k^2. \end{cases} \quad (10)$$

The coordinates of selected particles are generated using Gaussian distribution with estimated average  $\mu_x(t_m), \mu_y(t_m)$  and standard deviations  $\sigma_x(t_m), \sigma_y(t_m)$ . According to obtained coordinates the histogram  $q_i^{(j)}$  for each particle is determined. The estimation of the histograms and positions for generated particles  $H(e_2'')$ :

$$\forall d_i \in D(t_m) \mid d_i(t_m + 1) = \{x_i(t_m), y_i(t_m), q_i^{(j)}(t_m)\}, \quad (11)$$

where

$$\begin{aligned} x_i(t_m) &\sim \text{Gaussian}(\mu_x(t_m), \sigma_x(t_m)), \\ y_i(t_m) &\sim \text{Gaussian}(\mu_y(t_m), \sigma_y(t_m)), \\ w(e_3'', t_m + 1) &= t_m + \eta_k^3 \end{aligned} \quad (12)$$

After this event, the third internal event  $e_3''$  is invoked.

The comparison of histograms  $H(e_3'')$ :

For each particle the correlation coefficient  $\zeta_i$  is determined comparing its histogram  $q_i^{(j)}$  with the template histogram  $p^{(j)}$ .

$$\begin{cases} w(e_4'', t_m + 1) = t_m + \eta_k^4, \\ \zeta_i = \frac{\sum_{j=1}^N (q_i^{(j)} - \bar{q}_i)(p^{(j)} - \bar{p})}{\sqrt{\sum_{j=1}^N (q_i^{(j)} - \bar{q}_i)^2 \cdot \sum_{j=1}^N (p^{(j)} - \bar{p})^2}}, \\ \forall d_i \in D(t_m) \mid d_i(t_m + 1) = d_i(t_m) \cup \zeta_i. \end{cases} \quad (13)$$

All particles, which have a weight  $\zeta_i$  is equal or bigger than  $\beta$  are considered as proper and all others are eliminated from the set of particles  $D(t_m)$ . If in the set  $D(t_m)$  there are no particles which satisfy the weight requirements then the algorithm starts from the beginning, i.e., the generation of the new particles in the internal event  $e_1''$ . Otherwise, the internal event  $e_5''$  is invoked.

The selection of proper particles  $H(e_4'')$  is described below:

$$D(t_m + 1) = D(t_m) \setminus d_i, \quad d_i \in D(t_m); \quad (14)$$

$$d_i(t_m) = \{x_i(t_m), y_i(t_m), \zeta_i(t_m), q_i^{(j)}(t_m)\}, \quad \zeta_i(t_m) < \beta; \quad (15)$$

$$w(e_5'', t_m + 1) = t_m + \eta_k^5, \quad \text{if } D(t_m + 1) \neq \emptyset; \quad (16)$$

$$\begin{cases} w(e_1'', t_m + 1) = t_m + \eta_k^1, \\ \mu_x(t_m) = \infty, \\ \mu_y(t_m) = \infty, \end{cases} \quad \text{if } D(t_m + 1) = \emptyset. \quad (17)$$

In the case if the selected particles in the event  $e_4''$  are distributed in to the few separate classes, the maximum distance between particles is estimated. If this distance  $md$  exceeds  $\alpha$ , clustering  $e_6''$  is started. Otherwise, from the set of particles  $C(t_m)$  the new average coordinates are estimated and the location of tracking object is updated during internal event  $e_7''$ .

The estimation of max distance among particles  $H(e_5'')$ :

$$D(d_i, d_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \quad (18)$$

where  $md = \max(D(d_i, d_j))$ ,  $\max(md) = MAX(t_m)$ ,

$$D(d_1, d_2) = MAX(t_m), \text{ and}$$

$$w(e_6'', t_m + 1) = t_m + \mu_k^6, \quad \text{if } MAX(t_m) \geq \alpha. \quad (19)$$

$$\left. \begin{aligned} w(e_7^n, t_m + 1) &= t_m + \mu_k^7, \\ C(t_m) &= D(t_m). \end{aligned} \right\} otherwise \quad (20)$$

The selection of the proper group of particles using k-means clustering method  $H(e_6^n)$ :

For the grouping particles the clustering method is used. Only here, the initial centroids  $\bar{d}_1$  and  $\bar{d}_2$ , which are coordinates of most remote particles are determined in advance:

$$\bar{d}_1 = (\bar{x}_1, \bar{y}_1), \bar{d}_2 = (\bar{x}_2, \bar{y}_2). \quad (21)$$

Based on the  $\bar{d}_1$  and  $\bar{d}_2$ , distribution in to two groups  $C_1, C_2$  is performed. The distance from the particles  $\bar{d}_1$  and  $\bar{d}_2$  to the all others is estimated.

$$D(d_j, \bar{d}_k) = \sqrt{(x_j - \bar{x}_k)^2 + (y_j - \bar{y}_k)^2}, \quad j = \overline{1, n}, \quad k = \overline{1, 2}. \quad (22)$$

All particles are grouped into the two clusters.

$$\delta(d_j, \tilde{C}_k) = \begin{cases} 1, & D(d_j, \bar{d}_k) < D(d_j, \bar{d}_m); k, m = \overline{1, 2}; k \neq m; \\ 0, & otherwise. \end{cases} \quad (23)$$

Cluster, which has more particles, is selected for processing.

$$C(t_m) = \begin{cases} C_1, & \text{if } |C_1| > |C_2|, \\ C_2, & \text{if } |C_2| > |C_1|. \end{cases} \quad (24)$$

$$w(e_7^n, t_m + 1) = t_m + \mu_k^7. \quad (25)$$

After all actions the internal event  $e_7^n$  is invoked. During this event new average coordinates are estimated. The tracked position  $\mu_x(t_m), \mu_y(t_m)$  of the object is equal to the average coordinates of all particles from the set  $C(t_m)$ .

$H(e_7^n)$ :

$$\left\{ \begin{aligned} \mu_x(t_m) &= \frac{\sum_{j=1}^n \delta(d_j, C(t_m)) x_j}{|C(t_m)|}, \\ D(t_m + 1) &= \emptyset, \\ \mu_y(t_m) &= \frac{\sum_{j=1}^n \delta(d_j, C(t_m)) y_j}{|C(t_m)|}. \end{aligned} \right. \quad (26)$$

Since after this internal event the processing of the certain picture is finished, the output signal  $Y = y_1$  is generated.

The formal specification described above gives the requirements for the practical filter realization defining the algorithm as seven stages (internal events in specification).

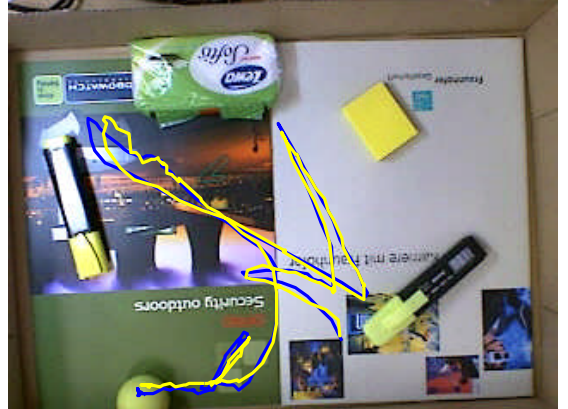
## Experiments

The randomly moving ball was taken on the video of 174 colored frames (size 640x480) in order to test the proposed dynamic particle filter. The complex scene with colored background and objects which have similar color

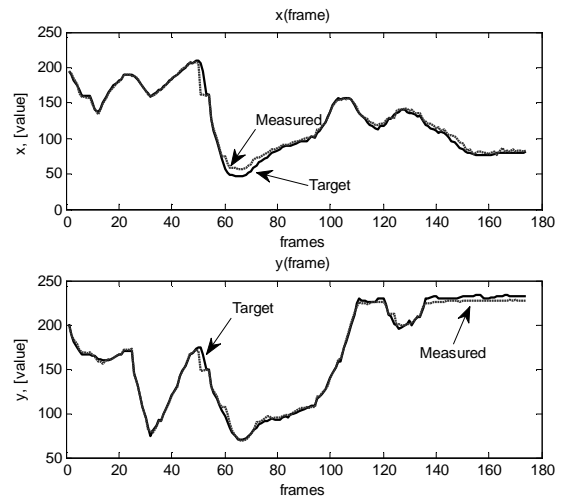
like target object was design for checking the robustness of the proposed approach. The video data was gathered with ordinary USB web-camera in the day light (point lightness about 1500 Lux). The template histogram of observable object was selected from the region of 30x30 pixels in the first frame. Starting position of the ball and the region size of the interest is selected by the user. Next positions of the ball are tracked using dynamic particle filter.

The test trajectory of the ball movement was designed measuring the real position of the ball in each frame by the hand. This trajectory was compared with the trajectory obtained by the proposed approach. The threshold of minimum distance between two ball positions in two different frames was changed from 0 to 5. Three numbers of particles (250, 500 and 3000) were used in the experiment. The mean execution time, mean absolute error between two trajectories and particle usage were measured during experiment. The tracking results are shown in Fig. 5–8.

The Fig. 4 shows the image of the experimental environment, target trajectory is colored in black and the measured trajectory is colored in white. The illumination of the background is not evenly distributed; therefore, measured trajectory is slightly different from the target trajectory in the places where more illumination is.



**Fig. 4.** The tracked (white) and test (black) trajectories of the ball in the complex scene

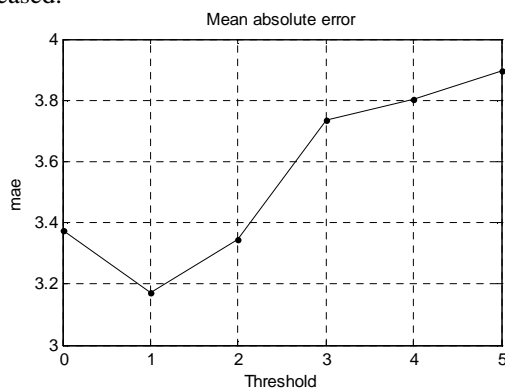


**Fig. 5.** The variation of the coordinates of tracked and test trajectories during the time

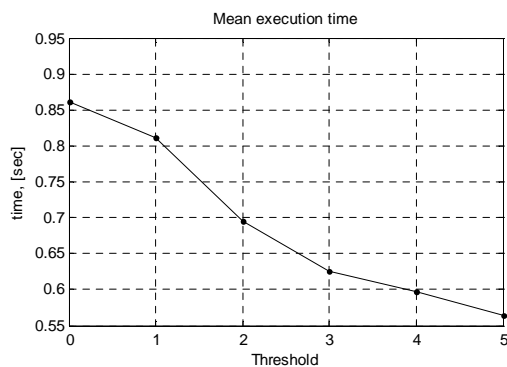
The coordinates of the target and measured trajectories extended in the time domain are shown in Fig. 5. It clearly visible, that the curves are similar. The proposed algorithm tracks the position of the observable object efficiently and robustly with relatively small offset (see Fig. 5).

The mean absolute error was computed between target and measured trajectories with different threshold values of the minimum distance. The curve is shown in Fig. 6. The MAE error increases, when the threshold value increases.

Mean execution time is the time, which is spent to track the position of the observable object in each video frame. The execution time was obtained using MATLAB, therefore, the comparison of results of similar experiments should consider this fact. The numerical relation between mean execution time and the threshold of the minimum distance is shown in Fig. 7. The curve shows, that the execution time decreases when the threshold value is increased.



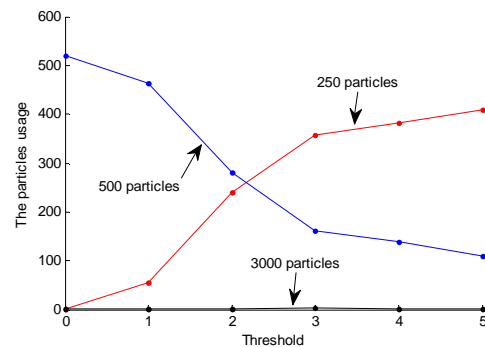
**Fig. 6.** The relation between the mean absolute error (*MAE*) variation and the threshold of the minimum distance. *MAE* is computed between test and tracked trajectories



**Fig. 7.** The relation between mean execution time of one video frame and the threshold of the minimum distance

More time (about 0.85 sec.) is needed to track the position of observable object when the threshold is equal to 0 and less (about 0.55 sec.) when the threshold is equal to 5. This can be explained with the relation between the particle usage and the threshold of the minimum distance. The particle usage shows how many times certain number of particles was used during the total video processing (see Fig. 7). Proposed algorithm uses 500 particles on each frame when the threshold of minimum distance is equal to 0. The algorithm uses 250 particles more frequently than 500 particles on each frame when the threshold is equal to 5. 3000 particles are used only in a few frames, when the

location of the observable object has changed rapidly from one frame to another. This number of particles is sufficient to cover almost full image and track the position of the lost object.



**Fig. 8.** The relation between the particles usage and the threshold of the minimum distance

Tracking algorithm based on dynamic particle filter dynamically changes the number of particles and allows to decrease the computation time. If less particles are used then the real position of the observable object can be found with certain error (see Fig. 6). However, the mean absolute error increases less then computation time decreases, accordingly, 19 % and 35 %.

## Conclusions and future works

The algorithm of the dynamic particle filter was presented, which enables the tracking of the non-rigid object in the complex background. The proposed algorithm was formalized using DPLA notation which allowed us to specify the dynamics of the real example in the mathematical, unambiguously and understandable manner. Experimental results show that the proposed dynamic particle filter can successfully follow the object when it moves in the complex background.

The future work will incorporate the extension of algorithm for automatic selection of the template histogram which takes into account the topological information, detection of suddenly appearing objects and multiple similar object tracking. The proposed algorithm can be used for the associative communication technologies, vision based security systems.

## Acknowledgments

The authors appreciate the financial support of the Lithuanian state science and studies foundation.

## References

1. **Forsyth D. A., Ponce J.** Computer vision; a modern approach // Englewood Cliffs, NJ: Prentice Hall. – 2002. – P. 235–240.
2. **Trucco E., Verri A.** Introductory techniques for 3D computer vision // Egkewiid Clifs, NJ: Prentice Hall, 1998. – P. 125–130.
3. **Paragios N., Chen Y., Faugeras O.** Handbook of mathematical models in computer vision // Springer. ISBN 0-387-26371. – 2006. – P. 293–307.

4. Vermaak J., Ikoma N., Godsill S. J. Sequential Monte Carlo framework for extended object tracking // IEE Proc.-Radar Sonar Navig. – 2005. – P. 152(2):353–363.
5. Shaohua Z., Chellappa R., Moghaddam B. Adaptive visual tracking and recognition using particle filters // ICME'03 Proc. 2003 Int. conf. on Multimedia and Expo. – 2003. – Vol. 2. – P. 349–352.
6. Shaohua Z., Chellappa R., Moghaddam B. Appearance tracking using adaptive models in a particle filter // Proc. of the 6<sup>th</sup> Asian conf. on Computer Vision. – 2004. – P. 86–90.
7. Martinez-del-Rincon J., Orrite-Urunuela C., Herrero-Haraba J. E. An efficient particle filter for color-based tracking in complex scene // Conf. on Advanced Video and Signal Based Surveillance. – 2007. – P. 176–181.
8. Barrero P., Canas M. J., Matellan V. Visual object tracking in 3D with color based particle filter // Proc. of World Academy of Science, Engineering and Technology. – 2005. – Vol. 4. – P. 200–203.
9. Isard M., Blake A. CONDENSATION – conditional density propagation for visual tracking // Int. Journal of Computer Vision. – 1998. – Vol. 20, No.1. – P. 5–28.
10. Rekleitis I., Dudek G., Milios E. Probabilistic cooperative localization and mapping in practice // IEEE Int. Conf. on Robotics and Automation (ICRA'03). – 2003. – Vol. 2. – P. 1907–1912.
11. Fujun P., Pingyuan C., Yangzhou C. Adaptive MCMC particle filter for nonlinear and non-Gaussian state estimation // Int. Conf. on Innovative Computing Information and Control (ICIC'08). – 2008. – P. 494–494.
12. Kurazume R., Yamada H., Murakami K., Iwashita Y., Hasegawa T. Target tracking using SIR and MCMC particle filters by multiple cameras and laser range finders // IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IRO'08). – 2008. – P. 3838–3844.
13. Thomas C. F., Phan L. L., Gerald S. R., Don M. T. Scalp electrode impedance, infection risk and EEG data quality // Clin Neurophysiol. – 2001. – P. 536–544.
14. Pranevičius H. Formal Specification and Analysis of Computer Network Protocols: Aggregate Approach // Kaunas: Technologija, 2004. – P. 50-180.
15. Pranevičius H., Makackas D., Paulauskaitė-Tarasevičienė A. Multi-Agent Systems Formalization Using Formal Method DPLA // The 21st annual European Simulation and Modelling Conference, St. Julians, Malta, 2007. – P. 427–431.
16. Paulinas M., Miniotas D., Meilūnas M., Ušinskas A. An Algorithm for Segmentation of Blood Vessels in Images // Int. Journal of Electronics and Electrical engineering. – 2008. – No. 3(83). – P. 25-28.
17. Bistrov V. Analyse of Kalman Algorithm for Different Movement Modes of Land Mobile Object // Int. Journal of Electronics and Electrical engineering. – 2008. – No. 6(86). – P. 89-92.
18. Laptik R., Navakauskas D. MAX-MIN Ant System in Image Processing // Int. Journal of Electronics and Electrical engineering. – 2009. – No. 1(89). – P. 21-24

Received 2009 01 28

**V. Raudonis, R. Simutis, A. Paulauskaitė-Tarasevičienė. An Efficient Object Tracking Algorithm based on Dynamic Particle Filter // Electronics and Electrical Engineering. – Kaunas: Technologija, 2009. – No. 3(91). – P. 93–98.**

The tracking problem of the non-rigid object has two difficulties. First, the object can change his scale, form or be partially occluded with other objects and, second, the location of the object can vary in the non-linear manner during the time. The paper considers the method, which solves given problem of the object tracking in the complex background. The solution of the mentioned problem is solved using dynamic particle filter (DPF), where the amount of the particles depends on the changing rate of the object location. The algorithm of dynamic particle filter was formalized using formal DPLA (dynamic PLA) notation. The object tracking is done according his color features (HSV histogram), which are independent from the object scale and/or form. Proposed method is demonstrated in the context of tracking real objects in the video data. The results shows, that the object tracking rate depends on the amount of generated particles and the tracking error. Ill. 8, bibl. 15 (in English; summaries in English, Russian and Lithuanian).

**V. Раудонис, Р. Симутис, А. Паулаускайте-Тарасевичене. Эффективный метод для отслеживания объекта, основанный на динамическом фильтре частиц // Электроника и электротехника. – Каунас: Технология, 2009. – № 3(91). – С. 93–98.**

Проблема отслеживания «нежестких» объектов имеет две трудности. Во-первых, объект может изменить свои масштабы, форму или быть частично за другими объектами, и, во-вторых, место расположения объекта может в течении времени варьироваться нелинейным образом. Рассматривается метод, который решает проблему слежения объекта в сложном фоне. Решить указанную задачу можно используя динамический фильтр частиц, в котором количество частиц зависит от меняющихся темпов местоположения объекта. Отслеживание объекта осуществляется по его цветным особенностям (HSV гистограмме), которые не зависят от масштаба объекта или формы. Предлагаемый метод демонстрируется при отслеживании реальных объектов в видео данных. Полученные результаты показывают, что продолжительность вычисления зависит от числа частиц. Ил. 8, библи. 15 (на английском языке; рефераты на английском, русском и литовском яз.).

**V. Raudonis, R. Simutis, A. Paulauskaitė-Tarasevičienė. Efektyvus objekto sekimo metodas, pagrįstas dinaminio dalelių filtru // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2009. – Nr. 3(91). – P. 93–98.**

„Nestandaus“ objekto sekimo uždavinys susiduria su dviem klausimais. Pirma, objektas vaizdo medžiagoje gali keisti savo formą, dydį arba būti iš dalies užstojamas, antra, objekto padėtis laiko atžvilgiu kinta netiesiškai. Straipsnyje pateikiamas dinaminis dalelių filtro algoritmas, kuris sprendžia objekto sekimo sudėtingos spalvinės sudėties fone uždavinį ir kurio dalelių skaičius priklauso nuo sekamo objekto padėties kitimo spartos. Algoritmas yra formaliai specifikuotas naudojant dinaminį PLA. Objektas vaizdo medžiagoje yra sekamas pagal jo spalvines savybes (HSV histogramą), kurios yra nepriklausomos nuo objekto dydžio ar formos. Dinaminio dalelių filtru galima sekti iš dalies užstotus ir „lanksčius“ objektus. Siūlomas algoritmas buvo aprašytas formaliai ir išbandytas su realiais vaizdo duomenimis. Metodo tyrimai parodė, kad objekto sekimo sparta yra tiesiai proporcinga generuojamam dalelių skaičiui ir sekimo tikslumui. Il. 8, bibl. 15 (anglų kalba; santraukos anglų, rusų ir lietuvių k.).