An Efficient Object Tracking Algorithm with Adaptive Prediction of Initial Searching Point*

Jiyan Pan, Bo Hu, and Jian Qiu Zhang

Dept. of E. E., Fudan University. 220 Handan Road, Shanghai 200433, P.R. China
{jiyanpan, bohu, jqzhang01}@fudan.edu.cn

Abstract. In object tracking, complex background frequently forms local maxima that tend to distract tracking algorithms from the real target. In order to reduce such risks, we utilize an adaptive Kalman filter to predict the initial searching point in the space of coordinate transform parameters so that both tracking reliability and computational simplicity is significantly improved. Our method tracks the changing rate of the transform parameters and makes prediction on future values of the transform parameters to determine the initial searching point. More importantly, noises in the Kalman filter are effectively estimated in our approach without any artificial assumption, which makes our method able to adapt to various target motions and searching step sizes without any manual intervention. Simulation results demonstrate the effectiveness of our algorithm.

Keywords: Object tracking, coordinate transform, initial searching point, adaptive Kalman filter.

1 Introduction

Object tracking has been widely applied to video retrieval, robotics control, traffic surveillance and homing technologies. A lot of object tracking algorithms have been reported in literatures, and among them the template matching algorithms has drawn much attention [1]-[6]. In such algorithms, target is modeled by a template, and is tracked in a video sequence by matching candidate image regions with the template through coordinate transforms. The set of transform parameters that yield the highest similarity between the template and the mapped image region of the current frame represents the geometric information of the target.

The performance of object tracking heavily depends on whether the search for the optimal transform parameters can be executed effectively. Many fast searching algorithms have been proposed in an effort to increase the accuracy of searching results

* This work is part of Research on Key Scientific and Technologic Problems of Molecular Imaging and is supported by National Basic Research Program 973 under Grant 2006CB705700.

L.-W. Chang, W.-N. Lie, and R. Chiang (Eds.): PSIVT 2006, LNCS 4319, pp. 1113 – 1122, 2006.

while reducing computational complexity. Typical algorithms include Three Step Search (TSS) [7], 2D-Log Search (2DLS) [8], Block-based Gradient Descent Search (BBGDS) [9], and Lucas-Kanade algorithm [1].

For all the algorithms mentioned above, the distraction of local minima is always a serious problem frequently leading to the failure to find the real coordinate transform parameters. Ideally, the image region where real target occupies in the current frame should render the largest similarity measure and therefore unambiguously make itself stand out against the other parts of the frame. When background is cluttered, however, some nearby objects also generate comparable similarity measure and hence confuse tracking algorithms. When searching for optimal coordinate transform parameters, tracking algorithms frequently find themselves trapped into local maxima produced by background objects and other interferences.

Such a situation can be improved by predicting the initial searching point in the space of transform parameters for the next frame and reducing searching range to ensure unimodalilty of the similarity measure. Since most local maxima in the transform parameter space reside some distance from the global maximum where the target locates, the risk of being trapped into local maxima can be substantially reduced if the initial searching point is in the close vicinity of the global maximum. This requires a good prediction of the geometric status of the target in each frame.

In the realm of object tracking, Kalman filters have been used in literatures [6], [11], [12], but few of them serve the purpose of predicting the initial searching point and enhancing tracking performance for the next frame. Besides, the model noises are fixed and determined empirically. In this paper, we propose an approach which employs Kalman filter to track the changing rate of the transform parameters instead of directly filtering their values. Then we select the predicted parameters as the initial searching point for the next frame. More importantly, after analyzing the cause of the model noises in the Kalman filter, we propose an effective method to estimate the power of those noises. As a result, the Kalman filter in our approach can automatically adapt to various target motions and searching step sizes. Experimental results indicate that the proposed method can achieve extremely high accuracy of predicting parameters and hence a significant decrease in the risk of being distracted by background interferences, as well as a considerable drop in computational burden.

The remainder of this paper is organized as follows. Section II focuses on the adaptive Kalman prediction of the initial searching point in the transform parameter space after a brief review of object tracking algorithms based on template matching. Experimental results are included in Section III. The paper is concluded in Section IV.

2 Adaptive Prediction of the Initial Searching Point

2.1 Object Tracking Based on Template Matching

The object (or target) to be tracked is characterized by an image called template which is generally extracted from the first frame of a video sequence. In subsequent frames of the video sequence, the template is mapped to the coordinate system of the frames by coordinate transforms. A searching algorithm tries various combinations of transform

parameters to find a set of transform parameters that maximize the similarity between the template and the mapped region of the current frame:

$$\mathbf{a}_m = \arg\max_{\mathbf{a}} \quad \mathrm{sim}\{I[\varphi(\mathbf{x};\mathbf{a})], T(\mathbf{x})\} \tag{1}$$

where $T(\boldsymbol{x})$ is the grey scale value of a template pixel located at $\boldsymbol{x}$ in the template coordinate system, $I(\boldsymbol{y})$ is the grey scale value of a frame pixel located at $\boldsymbol{y}$ in the frame coordinate system, $\varphi(\boldsymbol{x};\boldsymbol{a})$ is the coordinate transform with parameter vector $\boldsymbol{a}$, $\mathrm{sim}\{I,T\}$ is a function that measures the degree of similarity between images I and T. Typical examples of $\mathrm{sim}\{I,T\}$ include the normalized linear correlation or the inverse of SSD (sum of squared difference) between I and T [13]. $\boldsymbol{a_m}$ is the transform parameter vector that the searching algorithm assumes to be the one corresponding to correct geometric information of the target.

The type of the coordinate transform is determined by its parameter vector $\boldsymbol{a}$. For the coordinate transform that consists of translation, scaling and rotation, $\boldsymbol{a}$ has four components and $\varphi(\boldsymbol{x};\boldsymbol{a})$ can be written as

$$\varphi(\boldsymbol{x};\boldsymbol{a}) = a_1 \begin{bmatrix} \cos a_2 & \sin a_2 \\ -\sin a_2 & \cos a_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a_3 \\ a_4 \end{bmatrix} . \tag{2}$$

Generally speaking, $\varphi(\boldsymbol{x};\boldsymbol{a})$ can have arbitrarily large number of parameters and hence describe extremely complex object motions. Yet the model described by (2) is sufficient for most real-world tracking applications.

2.2 Predicting the Initial Searching Point

In order to predict the initial searching point in the transform parameter space, possible value of each transform parameter in the next frame has to be predicted. Since the frame rate is relatively high, we can reasonably assume the changing rate of each parameter does not alter abruptly over adjacent frame intervals. What brings uncertainty to the changing rate is the influence of arbitrary motion of the target. Such an influence brings about fluctuation of the changing rate of the transform parameters, and thus can be regarded as noise. We employ an adaptive Kalman filter to track the changing rate of the parameters. Such a method is especially instrumental in predicting, not just smoothing, the geometric status of the target. Since different transform parameters describe independent aspects of target motion, they can be predicted separately. The discussion below therefore focuses on one parameter alone and it can be applied to the other parameters trivially.

The state transition equation and the measurement equation for the changing rate of a coordinate transform parameter a are

$$v(n) = v(n-1) + u(n-1) , \tag{3}$$

$$v_m(n) = v(n) + w(n) , \tag{4}$$

where $v(n)$ is the changing rate of the parameter defined as $a(n)$-$a(n$-1), $v_m(n)$ is the measured changing rate of the parameter, which is actually the increment of the result of parameter search in (1), $u(n)$ is the cause of the fluctuation of $v(n)$ and is white with

the power of $\sigma_u^2(n)$, and $w(n)$ is the measurement noise resulting from the limit in the precision of the searching step size for the parameter a. It is also white, with the power of $\sigma_w^2(n)$.

Suppose $\hat{v}_P(n)$ is the prediction of v after the measurement up to frame n-1 is available, and $\hat{v}_E(n)$ is the estimate of v after the measurement up to frame n is acquired. If $e_P(n)$ denotes the prediction error of v and $e_E(n)$ represents the estimation error of v, the following equations hold:

$$v(n-1)=\hat{v}_E(n-1)+e_E(n-1)\ , \tag{5}$$

$$v(n)=\hat{v}_P(n)+e_P(n)\ . \tag{6}$$

Since the state transition coefficient in (3) is one, the estimate of v at frame n-1 serves as the prediction of v at frame n:

$$\hat{v}_P(n)=\hat{v}_E(n-1)\ . \tag{7}$$

From (3), and (5) to (7), the relationship between the prediction and the estimation errors can be derived:

$$e_P(n)=e_E(n-1)+u(n-1)\ . \tag{8}$$

As $e_E(n\text{-}1)$ is uncorrelated with $u(n\text{-}1)$, the additive relationship remains for the power of the signals in (8):

$$\sigma_P^2(n)=\sigma_E^2(n-1)+\sigma_u^2(n-1) \tag{9}$$

Where σ_P^2 and σ_E^2 are the power of prediction error and estimation error, respectively.

According to the theory of Kalman filtering [10], the optimal Kalman gain can be expressed as

$$G(n)=\frac{1}{1+\sigma_w^2(n)/\sigma_P^2(n)} \tag{10}$$

where the increase in the prediction error or the decrease in the measurement noise will lead to the rise in the Kalman gain.

After the measured value of v is obtained at frame n, the estimated value of it can be calculated using its predicted value and the Kalman-gain-weighted innovation:

$$\begin{aligned}\hat{v}_E(n)&=\hat{v}_P(n)+G(n)[v_m(n)-\hat{v}_P(n)]\\&=\hat{v}_P(n)+G(n)\alpha(n)\end{aligned} \tag{11}$$

where $\alpha(n)=v_m(n)-\hat{v}_P(n)$ is the innovation at frame n.

Updating the estimate of v leads to the renewal of estimation error as

$$\sigma_E^2(n)=[1-G(n)]\sigma_P^2(n). \tag{12}$$

(7) and (9) to (12) form a complete iteration to update the prediction of v.

After the predicted value of v for frame n+1 is obtained by applying (7) after (11), the prediction of a at frame n+1 can be written as

$$\hat{a}_P(n+1) = a_m(n) + \hat{v}_P(n+1) \tag{13}$$

where $\hat{a}_P(n+1)$ is the prediction of a at frame n+1, and $a_m(n)$ is the searching result of a at frame n. $\hat{a}_P(n+1)$ is usually very close to the real value of $a(n+1)$ and the initial searching point for a is therefore selected as $\hat{a}_P(n+1)$.

2.3 Estimating the Power of the Model Noises

Although the equations listed above seem to have solved our problem, the power of the two model noises, $\sigma_u^2(n)$ and $\sigma_w^2(n)$, remain to be estimated. Correct evaluation of them plays a key role in obtaining a proper Kalman gain and thus directly determines the performance of the Kalman filter. In the remainder of this section we would like to describe our approach to estimate $\sigma_u^2(n)$ and $\sigma_w^2(n)$.

As is mentioned before, the measurement noise is caused by the non-infinitesimal searching step size in looking for the optimal coordinate transform parameters. For simplicity of notation, we denote $a(n)$ as a_n. Suppose the step size for searching the parameter a_n is Δ, and the searching result is $a_{m,n}$. It is reasonable to assume that the true value of a_n is uniformly distributed over an interval of Δ centered at $a_{m,n}$; that is, the density of the true value of a_n is

$$p_n(a_n) = \begin{cases} 1/\Delta, & |a_n - a_{m,n}| \le \Delta/2 \\ 0, & elsewhere \end{cases}. \tag{14}$$

The power of searching error of a_n can be expressed as follows:

$$\begin{aligned} \sigma_a^2 &= \mathrm{E}\left\{(a_{m,n} - a_n)^2\right\} \\ &= \int_{-\infty}^{\infty} (a_{m,n} - a_n)^2 \, p_n(a_n) da_n \\ &= \int_{a_{m,n}-\Delta/2}^{a_{m,n}+\Delta/2} (a_{m,n} - a_n)^2 \frac{1}{\Delta} da_n \\ &= \frac{\Delta^2}{12} \end{aligned} \tag{15}$$

Since $v(n)$ is the changing rate of a_n, it is evident that

$$v(n) = a_n - a_{n-1}, \tag{16}$$

$$v_m(n) = a_{m,n} - a_{m,n-1}. \tag{17}$$

Taking (4), (16) and (17) into consideration, we can derive the power of measurement noise $\sigma_w^2(n)$ as follows:

$$\begin{aligned} \sigma_w^2 &= \mathrm{E}\left\{w^2(n)\right\} = \mathrm{E}\left\{(v_m(n) - v(n))^2\right\} \\ &= \mathrm{E}\left\{(a_{m,n} - a_n)^2\right\} + \mathrm{E}\left\{(a_{m,n-1} - a_{n-1})^2\right\} \\ &\quad - 2\mathrm{E}\left\{(a_{m,n} - a_n)(a_{m,n-1} - a_{n-1})\right\} \end{aligned} \tag{18}$$

As the parameter searching processes at different frames are uncorrelated, the cross term of (18) is zero. Considering (15), we can reduce (18) to

$$\sigma_w^2 = \mathrm{E}\left\{\left(a_{m,n} - a_n\right)^2\right\} + \mathrm{E}\left\{\left(a_{m,n-1} - a_{n-1}\right)^2\right\} = \frac{\Delta^2}{12} + \frac{\Delta^2}{12} = \frac{\Delta^2}{6} \ . \tag{19}$$

From (19) we can infer that having a finer searching step size can reduce the power of the measurement noise, which is just as expected.

The estimation of $\sigma_u^2(n)$, however, is not as straightforward since the motion of the target can be arbitrary. Yet we can still acquire its approximate value by evaluating the power of the innovation $\alpha(n)$. Considering (3), (4), (5) and (7) simultaneously, one can immediately get the following equation which relates the innovation with the estimation error and the two model noises:

$$\alpha(n) = e_E(n-1) + u(n-1) + w(n) \ . \tag{20}$$

The uncorrelatedness among the right-hand terms in (20) yields

$$\sigma_\alpha^2(n) = \sigma_E^2(n-1) + \sigma_u^2(n-1) + \sigma_w^2(n) \tag{21}$$

where $\sigma_\alpha^2(n)$ is the power of the innovation and can be approximated as

$$\sigma_\alpha^2(n) = \frac{1}{N} \sum_{k=n-N+1}^{n} \left[v_m(k) - \hat{v}_P(k)\right]^2 \ . \tag{22}$$

N is the number of frames over which the power of the innovation is averaged to obtain its approximate expectation.

Combining (19), (21) and (22), we can acquire the estimation of $\sigma_u^2(n)$ as follows:

$$\sigma_u^2(n) = \frac{1}{N} \sum_{k=n-N+2}^{n+1} \left[v_m(k) - \hat{v}_P(k)\right]^2 - \sigma_E^2(n) - \frac{\Delta^2}{6} \tag{23}$$

Where $\sigma_E^2(n)$ is calculated online in the iterations of Kalman filtering.

So far we have derived the expressions for estimating the power of the two model noises. By doing so, we do not have to assign any empirical values to those noises as most conventional approaches do. As a result, the method we have proposed can be applied to video sequences with various characteristics of target motion and searching algorithms with different searching step sizes, without any need to tune the Kalman filter manually.

The last step remaining is to initialize the filter. Since we have no information regarding target motion at the very beginning, it is natural to set the initial values of both $\hat{v}_E$ and σ_E^2 to be zero:

$$\hat{v}_E(0) = 0 \ , \ \sigma_E^2(0) = 0 \ . \tag{24}$$

3 Experimental Results

In order to examine how the adaptive prediction of the initial searching point in the transform parameter space can improve the performance of object tracking, we compare the tracking results of two algorithms that are exactly the same in every other aspect except that the first algorithm selects the transform parameters predicted by our

proposed method as the initial searching point for the next frame, and the other algorithm just takes the parameters found in the current frame as the initial searching point for the next frame. For simplicity, we denote the algorithm with adaptive prediction of initial searching point as Algorithm 1, and the other one is represented by Algorithm 2. The model of object motion includes translation and scaling. In both algorithms, the searching step size is 1 pixel for horizontal location and vertical location, and 0.05 for scale. Both algorithms select the inverse of SSD as the similarity function [2], and use gradient descent search algorithm to look for optimal transform parameters. Adaptive Kalman appearance filter is employed to update the template.

Figs. 1 to 3 illustrate how well our proposed method predicts the coordinate transform parameters in the next frame. We apply Algorithm 1 to a video sequence where the target undergoes much motion both in spatial locations and scales. Both actual and predicted values of the coordinate transform parameters for every frame are plotted in the same figure.

It can be seen from the figures that our method gives a very precise prediction of what the parameters are going to be in the next frame. The average distance between the initial searching point and the actual point in transform parameter space reduces from 2.7398 to 0.9632 when we use Algorithm 1 instead of Algorithm 2. Such a significant drop in the searching distance is extremely beneficial to tracking algorithms in terms of enhancing tracking stability and decreasing computational burden, as will be demonstrated in the following experimental results.

Fig. 4 and Fig. 5 exemplify considerable improvement of tracking stability when using the adaptive prediction of the initial searching point. When the initial searching point is much closer to the actual point in transform parameter space, tracking algorithms are less likely to be distracted by local maxima resulting from cluttered background, similar objects, or other interferences. This fact is confirmed by our experiments in which we deliberately choose a video sequence that has a vehicle running on a dark road at night. Due to the darkness, the vehicle is blurred and is somewhat similar to the road. When we apply Algorithm 2 to track the vehicle, it is not long before the algorithm loses the target because of being distracted by interferences from the road, as is shown in Fig. 4. Algorithm 1, however, successfully locks on the target throughout the sequence as is demonstrated in Fig. 5. The region in the lower right corner of each frame is the overlapped template.

Computational burden can also be greatly saved by the adaptive prediction of the initial searching point. Since the distance between the initial searching point and the final result point is substantially reduced, it takes searching algorithms in (1) much fewer trials to reach a final status, and computational complexity is therefore considerably reduced. Fig. 6 shows the parameter searching trial times of both algorithms. The right chart of Fig. 6 demonstrates the case where target has relatively high motion. The saving of computational burden is as high as 66.8%. Even in the case where target has low motion that is illustrated in the left chart of Fig. 6, Algorithm 1 can still lower computational complexity by 27.1%.

Since only scalar calculations are involved in the adaptive Kalman prediction of the initial searching point, the proposed algorithm can be implemented real time at a rate of 30fps using C codes on a Pentium-4 1.7GHz PC.

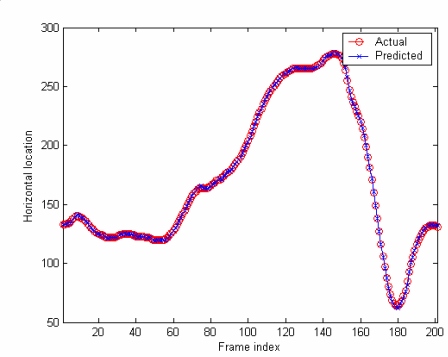

Fig. 1. Curves of the horizontal location of the target. The curve with circles represents the actual horizontal target location of every frame, and the curve with crosses depicts the predicted horizontal target location before every new frame is input.

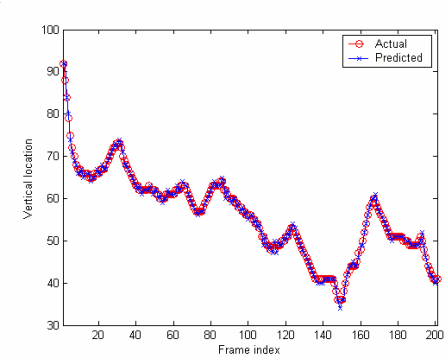

Fig. 2. Curves of the vertical location of the target. The meanings of different types of curves are the same as in Fig. 1.

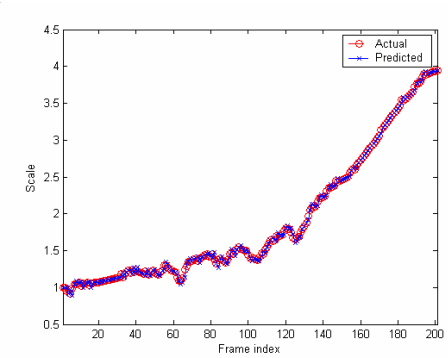

Fig. 3. Curves of the target scale. The meanings of different types of curves are the same as in Fig. 1.

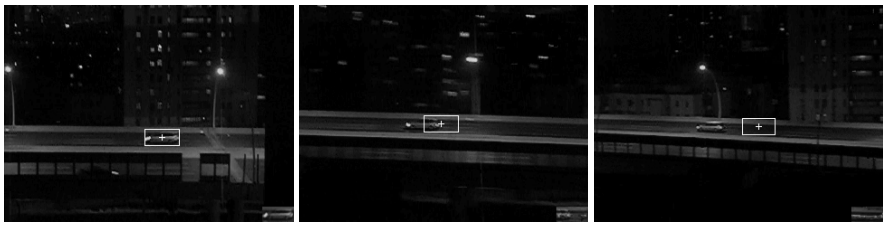

Fig. 4. Algorithm 2 fails to keep track of the vehicle when facing strong interferences from the background. Frame 1, frame 23 and frame 50 are displayed from left to right.

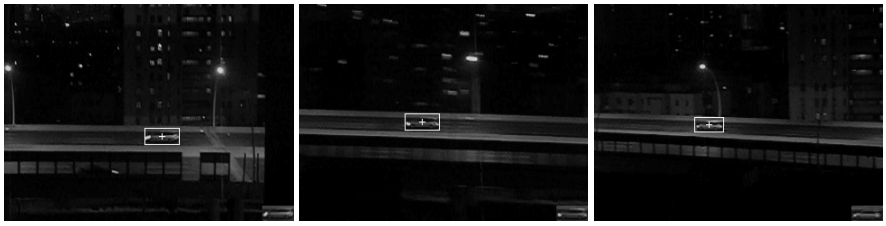

Fig. 5. Algorithm 1 tracks the vehicle perfectly all the time in spite of the existence of strong interferences from the background. Frame 1, frame 23 and frame 50 are displayed from left to right.

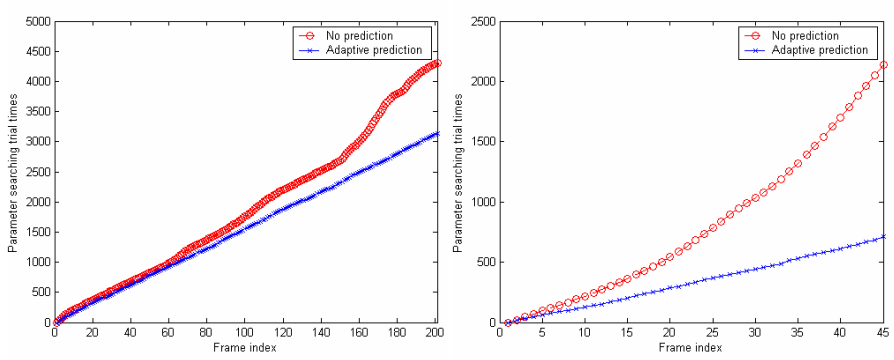

Fig. 6. Curves of parameter searching trial times over frame indices. The curves with circles show the result of Algorithm 2, and the curves with crosses illustrate the result of Algorithm 1. The left chart demonstrates the case where target has low motion, and the right chart, high motion.

4 Conclusion

In this paper we propose an algorithm which adaptively predicts possible coordinate transform parameters for the next frame and selects them as the initial searching point when looking for the real transform parameters. By doing so, tracking algorithms have less risk of being distracted by local maxima resulting from interferences, and tracking

performance is thus improved. We use an adaptive Kalman filter to achieve this purpose, but instead of directly filtering the values of transform parameters, we apply the Kalman filter on the changing rate of those parameters to effectively predict their future values. Moreover, we quantitatively analyze the cause of the model noises in the Kalman filter and derive their analytical expressions, so that the Kalman filter in our algorithm is automatically and correctly tuned when the characteristics of target motion change over time, or the searching algorithm uses different searching step sizes. Experimental results show that our proposed algorithm considerably promotes tracking stability while substantially decreasing computational complexity.

References

1. Baker S., Matthews I.: Lucas-Kanade 20 Years on: A Unifying Framework. Int'l J. Computer Vision, Vol. 53, No. 3. (2004) 221-255
2. Matthews I., Ishikawa T., Baker S.: The Template Update Problem. IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 26, No. 6. (2004) 810-815
3. Kaneko T., Hori Osamu: Template Update Criterion for Template Matching of Image Sequences. Proc. IEEE Int'l Conf. Pattern Recognition, Vol. 2. (2002) 1-5
4. Peacock A.M., Matsunaga S., Renshaw D., Hannah J.: A. Murray: Reference Block Updating When Tracking with Block Matching Algorithm. Electronic Letters, Vol. 36. (2000) 309-310
5. Smith C., Richards C., Brandt S., Papanikolopoulos N.: Visual Tracking for Intelligent Vehicle-highway Systems. IEEE Trans. on Vehicle Technology, (1996.)
6. Papanikolopoulos N., Khosla P., and Kanade T.: Visual Tracking of a Moving Target by a Camera Mounted on a Robot: A Combination of Control and Vision. IEEE Trans. on Robotics and Automation, Vol. 9. (1993) 14-35
7. Wang Y., Ostermann J., Zhang Y.Q.: Video Processing and Communications. Prentice Hall (2002) 159-161
8. Jain J. Jain A.: Displacement Measurement and Its Application in Interframe Image Coding. IEEE Trans. on Communications, Vol. 33. (1981) 1799-1808
9. Liu L.K., Feig E.: A block-based gradient descent search algorithm for block motion estimation in video coding. IEEE Trans. on Circuits and Systems for Video Technology, Vol. 6. (1996) 419-422
10. Brown R.G. and Hwang P.Y.C.: Introduction to Random Signals and Applied Kalman Filtering. John Wiley (1992)
11. Blake A., Curwen R., Zisserman A.: A Framework for Spatio-Temporal Control in the Tracking of Visual Contour. Int'l J. Computer Vision, Vol. 11, No. 2. (1993) 127-145
12. Isard M., Blake A.: CONDENSATION – Conditional Density Propagation for Visual Tracking. Int'l J. Computer Vision, Vol.29, No. 1. (1998) 5-28
13. Sezgin M., Birecik S., Demir D., Bucak I.O., Cetin S., Kurugollu F.: A Comparison of Visual Target Tracking Method in Noisy Environments. Proc. IEEE Int'l Conf. IECON, Vol. 2. (1995) 1360-1365