Robust Moving Object Detection from a Moving Video Camera Using Neural Network and Kalman Filter

Sanaz Taleghani[1], Siavash Aslani[2], and Saeed Shiry[3]

[1] Mechatronics Research Laboratory (MRL), Department of Computer & IT, Qazvin Azad University, Qazvin, Iran
taleghani@mrl.ir
[2] Department of Computer & IT, Qazvin Azad University, Qazvin, Iran
saslani@mrl.ir
[3] Computer Engineering Department, Amirkabir University, Tehran, Iran
shiry@aut.ac.ir

Abstract. Detecting motion of objects in images, while the camera is moving, is a complicated task. In this paper, we propose a novel method to effectively solve this problem by using Neural Network and Kalman Filter. This technique uses parameters of camera motion to overcome problems caused by error in the image processing outputs. We have implemented this technique in the MRL Middle Size Soccer Robots. The experimental results show a low error rate of 2.2% which suggests that the combined approach performs significantly better than the traditional techniques.

Keywords: Motion Detection, Neural Network, Kalman Filter, Middle Size Soccer Robot.

1 Introduction

For many applications of autonomous robots it is important to detect motion of objects that are in the surroundings of the robot to avoid collisions or enable interaction. The motion detection methods that are based on image processing need high quality images as input. Since the camera used to record images for a soccer playing robot is moving, the quality of images decreases and vibration of camera produces more fatal noise. This causes the output of image processing to include considerable error which makes feature detection and object recognition a complicated task.

Various filters would help to reduce the noise to some extent. Regarding moving objects tracking, Kalman Filtering, Extended Kalman Filtering and Particle Filtering (also known as Condensation and Monte Carlo algorithms) are some of the most common used algorithms. Kalman Filter provides an efficient recursive solution which has a prediction and correction mechanism, in a way that minimizes the mean of the squared error. Due to its simplicity, Kalman filter is still being used in most of the general-purpose applications [1].

L. Iocchi et al. (Eds.): RoboCup 2008, LNAI 5399, pp. 638–648, 2009.

For localization of moving objects, Nordlund introduced a method to detect objects motion using a sequence of images which indicates the object position after two frames, but the result improves over time [2]. Such approaches are based on difference of images for motion detection, but they have some difficulties with slow-moving objects. To solve this problem, some approaches employ techniques based on estimation such as Kalman Filter and neural networks [3].

Neural networks have been implemented for image tracking applications, where they are used mostly as classifiers or measures between different types of filters [4]. Zhang and Minai [5] created a two layer pulse coupled neural network for motion detection. The two layers work in iterative fashion and find the largest matching segment between two consecutive video frames. This model adopts the image pixels as the local feature. Based on Grossberg’s spreading theory and Ullman’s motion decision theory [6], Guo Lei proposed a spreading and concentrating model [7] to perform motion detection. The local feature used for motion detection is the edge elements in the object’s contour. The common problem of these models is the model complexity [8].

In this research we took benefits of neural networks in their learning ability and noise immunity. A MLP neural network was embedded in the Kalman Filter cycle as a corrector section. The result of implementation is quite satisfactory.

2 Problem Statement

Motion detection using image processing while the camera is moving is a difficult task. In such conditions, images are not clear and vibration of the camera decreases the quality of images significantly and causes high error in the image processing result. In the domain of soccer playing autonomous robots, it is important to have knowledge of the ball position and movement. Middle size soccer robots are a sample of systems that motion detection has an especial importance in their decision making.

As an appropriate test-bed, we have used MRL team's middle size soccer robots to implement and test different methods for detecting ball motion. These robots are

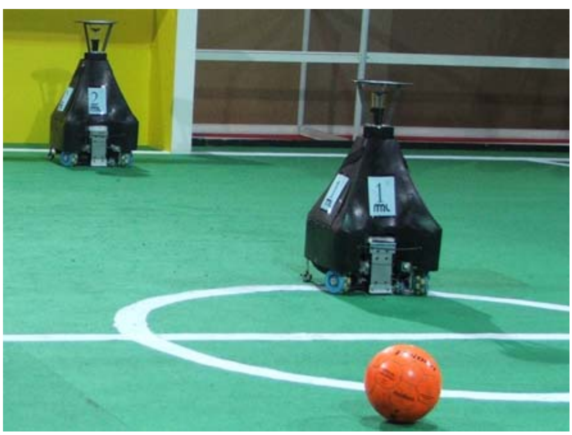

Fig. 1. MRL middle size soccer robot

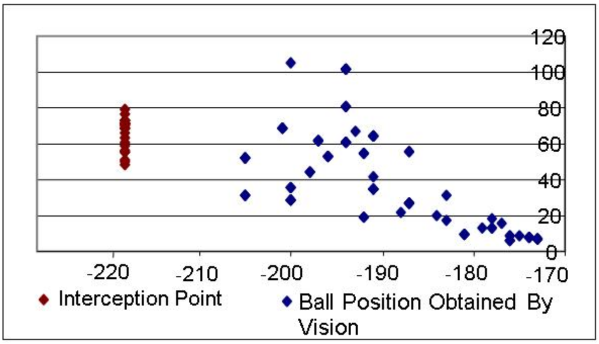

Fig. 2. The illustration of ball interception problem. Although the ball is stationary but robot supposes that the ball is moving because of the vision error in ball positioning. Robot tries to intercept the ball in the points shown in red.

equipped with omni-directional vision which consists of a camera and a hyperbolic mirror on top of each robot (Fig. 1). This kind of camera assemblies requires the lens to be mounted exactly in the focal point of the mirror while the camera has to be aligned with its symmetry axis. Due to the forces exposed to the robot during a game the camera will encounter considerable vibrations.

Movement and vibration rate of the robots during the game is significantly high which causes a considerable noise in information provided by the vision system. This noise makes the vision system to calculate different positions for a fixed ball within sequential steps. It leads the robot to try to calculate ball path and intercept it while it is not moving at all (Fig. 2).

3 Motion Detection

Having examined a number of traditional techniques of motion detection we realized they are not capable of providing acceptable result for the abovementioned test-bed. Therefore we tended to come up with a novel and optimized heuristic method.

This section explains the mentioned methods and results of their implementation in MRL middle size soccer robots.

3.1 Average and Threshold Method

This approach has a simple mathematical base. The position of objects that is observed by the vision system is recorded in some sequential steps. The difference between two successive object positions shows the movement value during the time. On the other hand each individual position of objects per steps is not precise because of the existing error in vision system. Using the discrepancy between average of some initial data calculated in (1) and average of some ultimate data calculated in (2) would be enough to resolve this problem. Thus if this discrepancy d in (3) is greater than a threshold value, we may assume that the object is moving, if otherwise then we would assume that the object is standstill.

$$\bar{X}_f = \frac{\sum_{i=1}^{n} x_i}{n} \quad \bar{Y}_f = \frac{\sum_{i=1}^{n} y_i}{n} \tag{1}$$

$$\bar{X}_l = \frac{\sum_{i=l-n}^{l} x_i}{n} \quad \bar{Y}_l = \frac{\sum_{i=l-n}^{l} y_i}{n} \tag{2}$$

$$d = \sqrt{(\bar{X}_f - \bar{X}_l)^2 + (\bar{Y}_f - \bar{Y}_l)^2} \tag{3}$$

In the above equations, X and Y are assumed as objects' coordinates in l sequential steps. In the implementation environment in middle-size robot, each time step lasts for 0.03 sec. After required information about ball position was registered in every short period of time (almost 0.5 sec) due to such data dispersal, the mentioned threshold could not be calculated for conditions which robot was moving (Fig. 3).

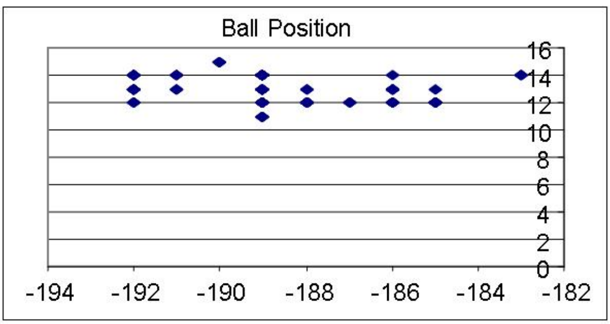

Fig. 3. Raw data observed by robot's vision. Ball position while both robot and ball are stationary.

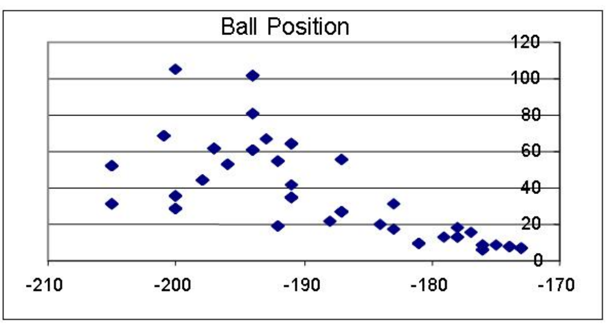

Fig. 4. Raw data observed by robot's vision. Ball position while robot is moving and the ball is stationary.

Fig. 4 presents the dispersal of ball position when ball is stationary but robot is moving. The dispersal depends on various speeds of robot which involves camera vibration and other environment error.

3.2 Kalman Filter

The Kalman Filter is a set of mathematical equations that provides an efficient computational recursive means to estimate the state of a process, in a way that minimizes the mean of the squared error [9].

The discrete Kalman Filter attempts to estimate the state of a discrete-time controlled process by using a form of feedback control. This means that merely the estimated state from the previous time step and the current actual measurement are required to compute the estimation for the current state. Thus equations for Kalman Filter can be divided into two groups: predictor equations and corrector equations:

- Predictor equations are responsible for projecting the current state estimate ahead in time.

$$\hat{x}_k^- = A\hat{x}_{k-1} + BU_{k-1} \tag{4}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{5}$$

$\hat{x}_k$ is defined as the estimate of the state at time k . P_k is defined as the error covariance matrix at time k that can measure accuracy of the estimated state. Q is the process noise covariance that might change with each step however here we define it as constant and also set U_k =0.

- The corrector equations are responsible for adjusting projected estimate by an actual (noisy) measurement at that time.

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \tag{6}$$

$$\hat{x}_k = \hat{x}_k^- + K_k (Z_k - H\hat{x}_k^-) \tag{7}$$

$$P_k = (1 - K_k H)P_k^- \tag{8}$$

K in (6) is defined to minimize the error covariance. R in (6) is the measurement noise covariance and is supposed to be constant. At this level of Kalman Filter algorithm $\hat{x}_k$ is obtained from a priori state estimate at step k and an actual measurement Z_k . The difference $Z_k - H\hat{x}_k^-$ in (7) reflects the discrepancy between an actual measurement Z_k and the predicted measurement $H\hat{x}_k^-$.

The ongoing Kalman Filter cycle is presented in Fig. 5. Indeed the Kalman Filter contains the different parts covering the high-level operation.

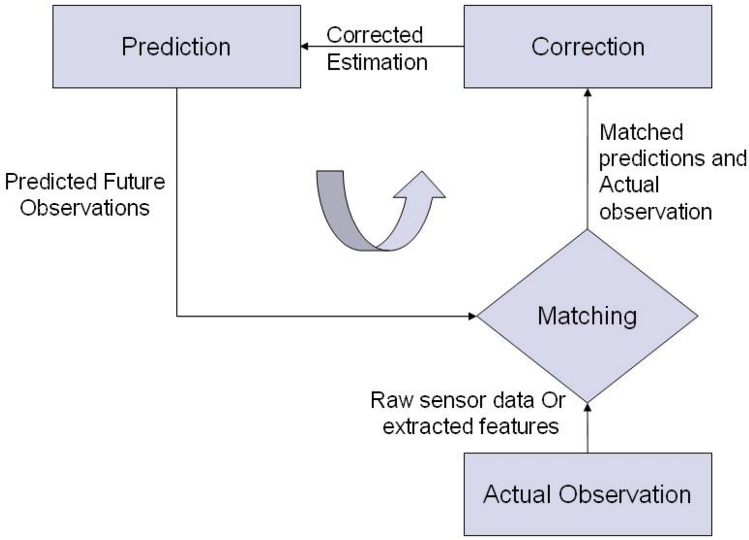

Fig. 5. The ongoing common Kalman Filter cycle

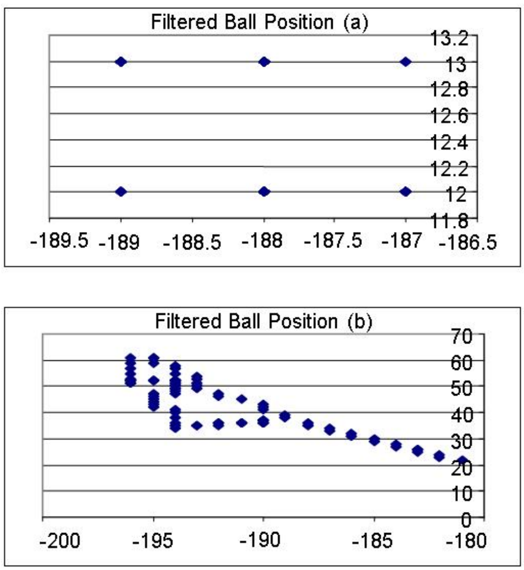

Fig. 6. Kalman Filter is not effective enough to correct ball position while the robot is moving. a) Kalman Filter reduced the noise of data of Fig 3. b) Kalman Filter could not reduce the noise of Fig 4.

In our implementation, the constant values for Kalman Filter equations are computed as Q=0.001 and R=0.5. We use ball position as output data of Kalman Filter which is provided by vision of robot. This approach is not able to recognize a fixed ball from a moving ball when the robot itself has movement in the field. Especially in the cases that the speed of robot is high, this approach may work appropriately. Fig. 6 clearly shows this issue.

3.3 Neural Network in Combination with Kalman Filter

The advantages of neural networks are twofold: learning ability and versatile mapping capabilities from input to output [10]. The multilayer perceptron is a nonparametric technique for performing a wide variety of estimation tasks [11].

Taking advantages of Artificial Neural Networks in learning and estimation, and combining it with Kalman Filter concept, we introduce an efficient technique to detect object motion. As shown in Fig. 5 Kalman Filter has different parts. In this approach, the Match and Estimation tasks of Kalman Filter are assigned to a MLP neural network that would be trained using backpropagation algorithm (Fig. 7).

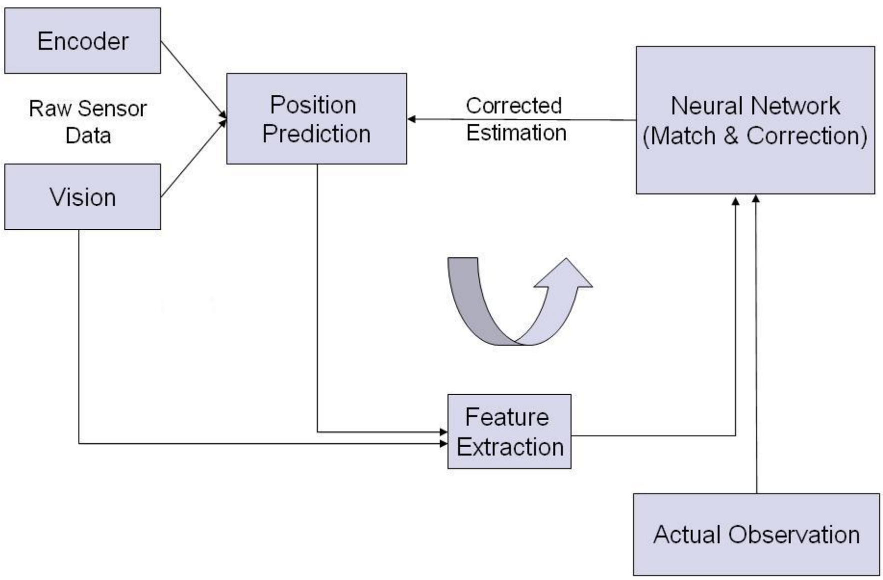

Fig. 7. The ongoing Kalman Filter cycle with embedded neural network

The location of object relative to a supervisor (A_1) and the supervisor's movement vector ΔO in serial steps are given as inputs to the system. Using these inputs, the relative location of the object in next step (A_2) can be predicted. Then the difference of the predicted value and the precept value of the vision system in next step is given to the neural network via multiplex sets. Since the training of neural network is supervised, the target value has to be supplied for every sample.

The addition of localization noise to the noise of relative ball position causes high error in global ball position. Therefore we use relative position of the ball instead of its global position. On the other hand in our vision system, the noise of Polar relative ball position is less than the noise of Cartesian global ball position. Thus, it is preferred to use Polar coordinates instead of Cartesian coordinates.

In our implementation in every step, the predictor part uses both of (r,θ) and $\Delta O\,(dx,dy,d\varphi)$. As shown in Fig. 8, at the beginning of a step the robot is positioned in the point A_1 and observes the ball in relative position B_1. During the step, robot moves as$(dx,dy,d\varphi)$ and positions in the point A_2 and observes the ball in relative position B_2 in next step.

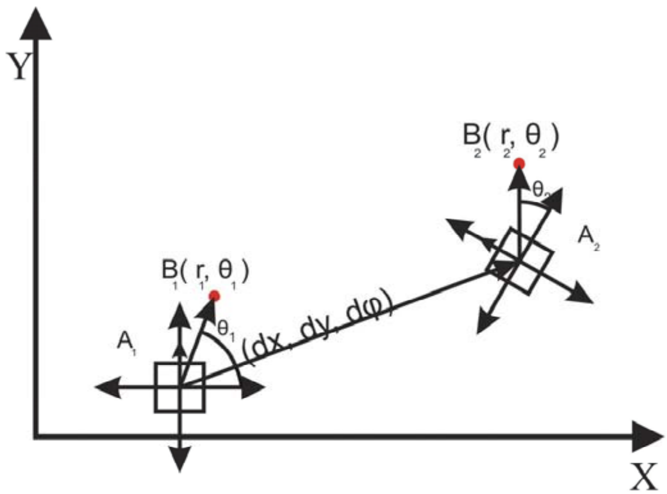

Fig. 8. At first the robot is placed at the point A_1 and observes the ball in the point B_1. Then it moves to the point A_2 and observes the ball in the point B_2 in next step.

We can compute the new relative position of the ball using movement vector of the robot according to the following equations.

$$A'_x = r_1 \cos\theta_1 - dx \tag{9}$$

$$A'_y = r_1 \sin\theta_1 - dy \tag{10}$$

$$B_x = A'_x \sin(d\phi) + A'_y \cos(d\phi) \tag{11}$$

$$B_y = A'_x \cos(d\phi) - A'_y \sin(d\phi) \tag{12}$$

$$(B_x, B_y) \xrightarrow{CartToPol} (r_2, \theta_2) \tag{13}$$

As predictor output is not accurate and the ball may also be moving, thus the predicted value differs from actual ball position observed by the vision in next step.

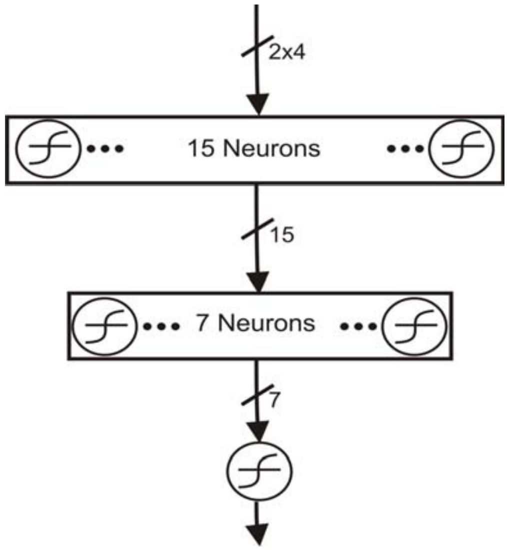

Fig. 9. Designed MLP Neural Network

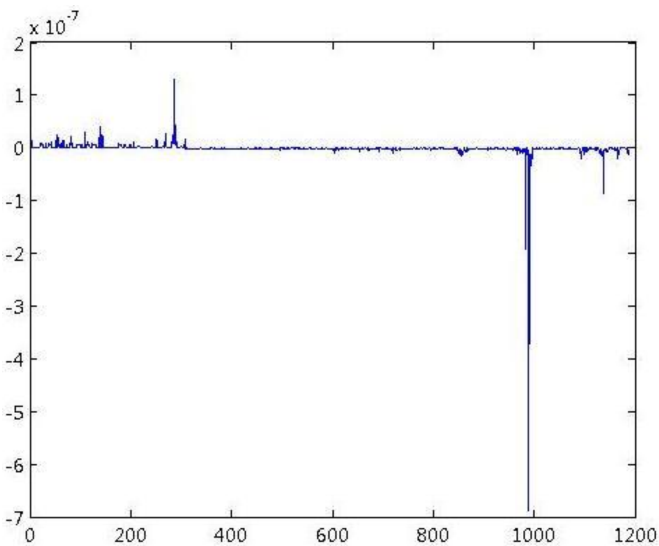

Fig. 10. Error in the training set with 1188 samples

The feature extraction section calculates this difference and after normalization passes it as input vectors to the neural network which is responsible for estimating and matching in Kalman Filter.

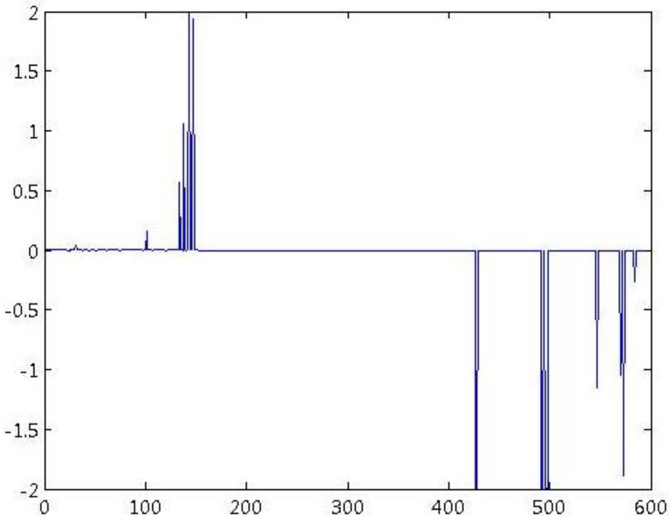

Fig. 11. Error in the validation set with 596 samples

We trained a 15-7-1 feedforward neural network by backpropagation algorithm. Neurons had *tansig* transition function (Fig 9).

The positive outputs were supposed to indicate moving ball and the negative outputs used to indicate stationary ball. The data of ball position and robot movement were logged in different states. The Robot moved in different speeds while the ball was stationary or moving in different directions. The collected data were split into a training set of 1188 vectors and a validation set of 594 vectors. During network training, it could learn all the training samples with great performance as shown in Fig 10.

Then the network was tested using the validation samples. Achieving 2.2% error shows that the trained network is not overfitted and is capable of detecting the ball motion in different states (Fig. 11).

4 Conclusion

As the movement of camera and its vibration leads to unclear and low quality noisy images, the image processing unit produces outputs with high error which makes the usual methods of motion detection unable to perform appropriately.

Having confronted with such problem we tended to design a novel technique that takes advantages of Neural Networks in learning nonlinear relations and combines it with Kalman Filter. As this technique is not directly involved with image processing procedure, it would be useful for systems where their input is provided by other resources and sensors. The result of implementation in MRL Middle Size Robots shows considerable performance improvement and suggests that the combined approach performs significantly better than traditional techniques.

References

1. Ruiz-del-Solar, J., Vallejos, P.A.: Motion Detection and tracking for an AIBO robot using camera motion compensation and Kalman Filtering. In: Robocup Symposium, Bremen (2006)
2. Nordlund, P.: Localization of a Moving Object by a Moving Observer. Technical report, Royal Institute of Technology, Stockholm (1994)
3. Herrero, E., Orrite, C., Alcolea, A., Roy, A., Guerrero, J.J., Sagues, C.: Video-Sensor for Detection and Tracking of Moving Objects. In: Perales, F.J., Campilho, A.C., Pérez, N., Sanfeliu, A. (eds.) IbPRIA 2003. LNCS, vol. 2652, pp. 346–353. Springer, Heidelberg (2003)
4. Spinko, V., Shi, D., Ng, W.S.: Endoscope tracking using Wavelet-Gravitation Network incorporated with Kalman Filter. In: 18th IEEE International Conference on Tools with Artificial Intelligence (2006)
5. Zhang, X., Minai, A.A.: Detecting corresponding segments across images using synchronizable Pulse-Coupled neural networks. In: IEEE/NNS International Joint Conference on Neural Networks, Washington, pp. 82–825 (2001)
6. Ullman, S.: The interpretation of visual motion. MIT Press, London (1979)
7. Lei, G., Baolong, G.: Visual system and distributed deduction theory. Xidan University Publisher (1995)
8. Yu, B., Zhang, L.: Pulse Coupled Neural Network for Motion Detection. In: IEEE International Joint Conference on Neural Networks, vol. 2, pp. 1179–1184 (July 2003)
9. Welch, G., Bishop, G.: An introduction to Kalman Filter. UNCChapel Hill (April 5, 2004)
10. Neji, Z., Beji, F.M.: Neural Network and time series identification and prediction. In: IEEE-INNS-ENNS International Joint Conference on Neural Networks (2000)
11. Werbos, P.J.: Beyond regression: new tools for prediction and analysis in the behavioral sciences. Ph.D. thesis, Harvard Univ., Cambridge, MA (1974)