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

Sanaz Taleghani[1], Siavash Aslani[2], 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 Networks and Kalman Filter. This technique uses movement parameters of camera to overcome problems caused by error in image processing outputs. We implemented this technique in the MRL Middle Size Soccer Robots. Experimental results are presented and 2.2% achieved error suggests that the combined approach performs significantly better than 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 moving 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 is moving, the quality of images decreases and quake of camera causes more fatal noise. This causes the output of image processing to include considerable error which makes many problems in the following computations.

Various filters would help to reduce 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, the Kalman filter is still been used in most of the general-purpose applications [1].

Neural networks have been implemented for image tracking applications, where they are used mostly as classifiers or measures between different types of filters [2]. Zhang and Minai [3] 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 [4], Guo Lei proposed a spreading and concentrating model [5] 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 [6].

In this research we took benefits of neural networks in their learning ability and noise immunity, whether a MLP neural network was embedded in Kalman Filter cycle as corrector section. Implementation result of this technique is too satisfactory and offers this combined technique to overcome such problems.

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 while the camera is quaking quality of images decreases significantly and causes high noise on image processing results. In the domain of soccer playing autonomous robots, it is important to have knowledge of the current 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 equipped with omni-directional vision which consisted 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 the robot is exposed during the game the camera is facing with vibration.

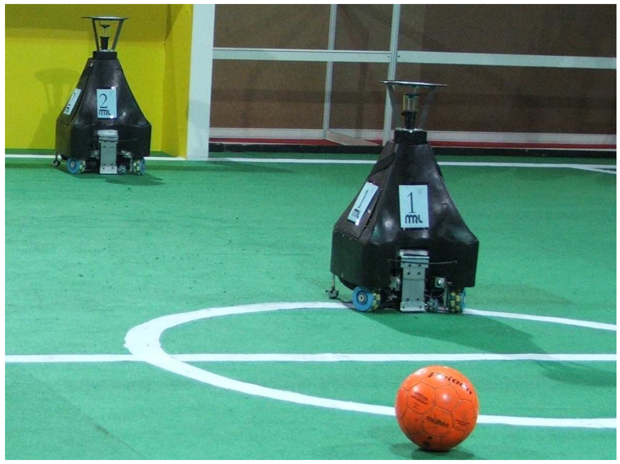

Fig. 1. MRL middle size soccer robot

Movement and quake rate of the mentioned robots during the game is significantly high and therefore causes a high noise in information provided by 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).

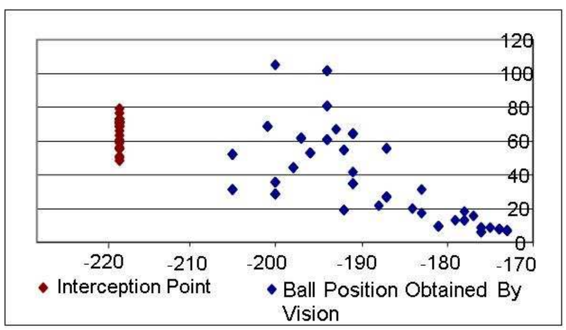

Fig. 2. The illustration of ball interception problem. Ball is stationary but robot supposes that ball is moving because of vision error in ball position. Robot tries to intercept ball in red points.

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 information of objects that would be observed by vision is recorded in some sequential steps. The difference between first object position and last object position shows the movement value during time. On the other hand each individual position of objects per steps is not precise because of the existing error in vision. 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.

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

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

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

In equations above, 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 needed 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 condition of which robot was moving (Fig. 3).

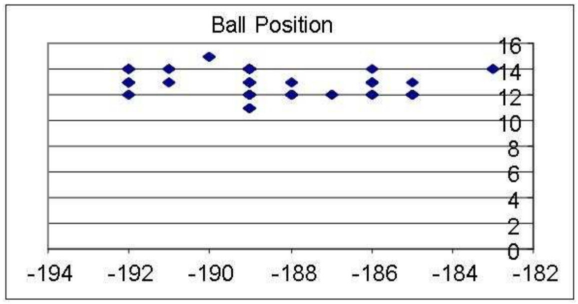

Fig. 3. Raw data observed by robot's vision. Ball position while both robot and ball are stationary.

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

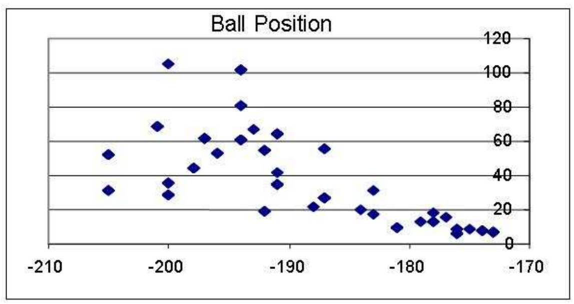

Fig. 4. Raw data observed by robot's vision. Ball position while robot is moving and ball is stationary.

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 [7].

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 estimated state. Q is the process noise covariance that might change with each step however here we define it is constant and also U_k=0.

- 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 measurement noise covariance supposed constant. At this level of Kalman filter algorithm $\hat{x}_k$ is obtained from a priori state estimate at step k and 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.

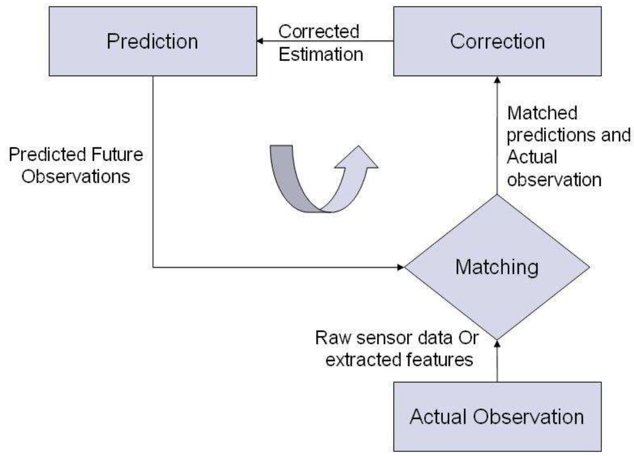

Fig. 5. The ongoing standard Kalman Filter cycle

In the implementation we began with goal of finding constant values of Kalman Filter equations that in our system 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 case that speed of robot is high, this approach does not work appropriately. Fig. 6 clearly shows this issue.

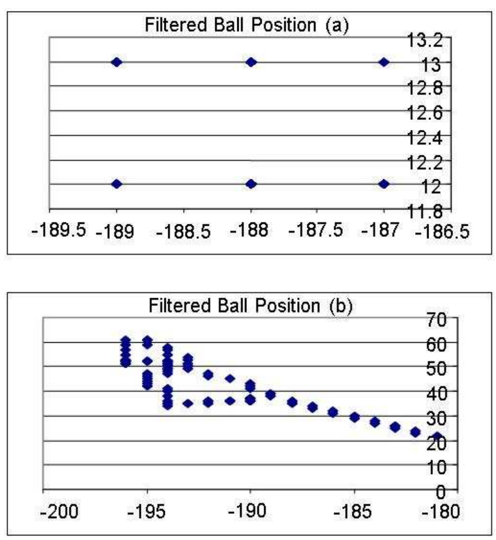

Fig. 6. Kalman filter is not effective enough to correct ball position while robot is moving. a) Kalman Filter reduced the noise of data of Fig 3. b) Kalman Filter could not reduce the noise of Fig 4.

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 [8]. The multilayer perceptron is a nonparametric technique for performing a wide variety of estimation tasks [9].

Taking advantages of Artificial Neural Networks in learning and estimation, and combining it with Kalman Filter conceptions, we introduce an efficient technique to detect objects' motion. As mentioned 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).

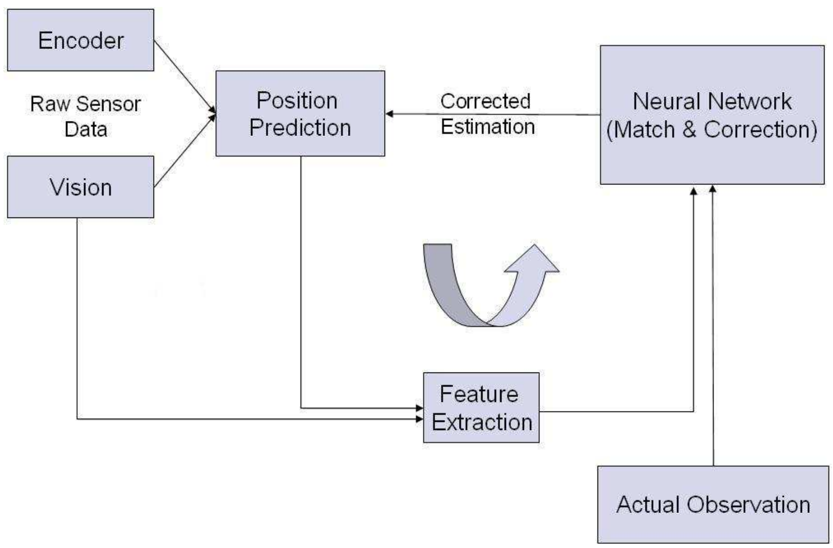

Fig. 7. The ongoing Kalman Filter cycle with embedded neural network

The location of object relative to supervisor (A_1) and the supervisor's relocation vector ΔO in serial steps are given as inputs to the system. Using these inputs the relative location of object in next step (A_2) can be predicted. Then 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 in this technique the training of neural network is supervised, the target value has to be supplied for every sample.

The addition of noise of localization to noise of relative ball position causes high error in global ball position. Therefore we used relative position of ball instead of its global position. On the other hand in MRL robots vision, the noise of θ in Polar relative ball position is less than the noise of r. Thus, use of Polar coordination is preferred to Cartesian coordination.

In the implementation, the pair (r,θ) and the relocation of robot during step $\Delta O\,(dx,dy,d\varphi)$ is used in predictor part. As shown in Fig. 8, at the beginning of the step robot is positioned in 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 point A_2 and observes the ball in relative position B_2 in next step.

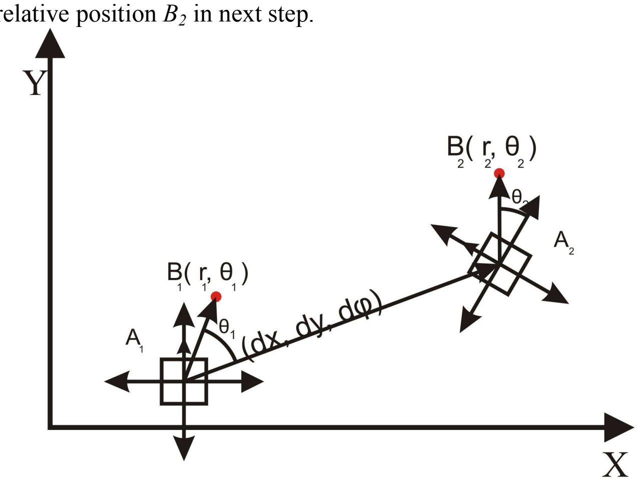

Fig. 8. First robot is placed at point A_1 and observes ball in point B_1. Then it moves to point A_2 and observes ball in point B_2 in next step.

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

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

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

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

$$B_y = A'_x \cos(d\phi) - A'_y \sin(d\phi) \tag{22}$$

$$(B_x, B_y) \xrightarrow{CartToPol} (r_2, \theta_2) \tag{33}$$

As predictor output is not accurate and ball may also be moving, thus predicted value differs from actual ball position observed by vision in next step. Feature extraction section calculates this difference and after normalization passes them as input vectors into the neural network which is responsible for estimating and matching in Kalman filter.

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

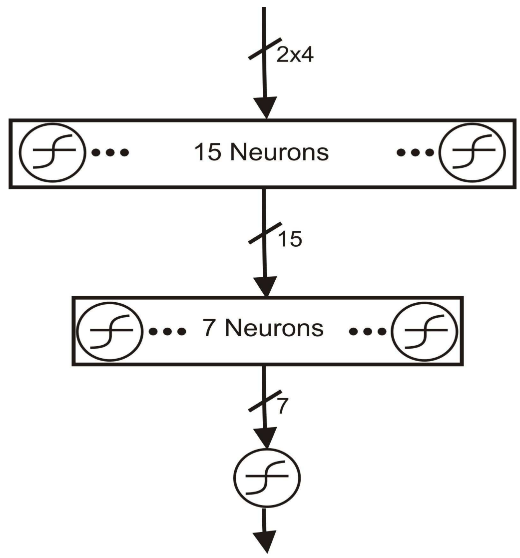

Fig. 9. Designed MLP Neural Network

In this case positive outputs were supposed to indicate moving ball and negative outputs used to indicate fixed ball. Ball position and robot's repositioning data was logged in different states. The Robot was used to move in different speeds while the ball was stationary or moving in different directions. The collected data was split into a training set of 1188 vectors and validation set of 594 vectors. Training the network, it learned all training samples with great performance as shown in Fig 10.

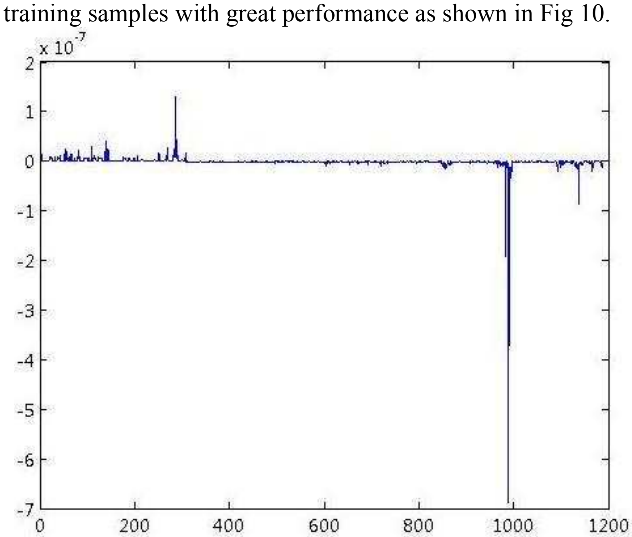

Fig. 10. Error in training set with 1188 samples

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

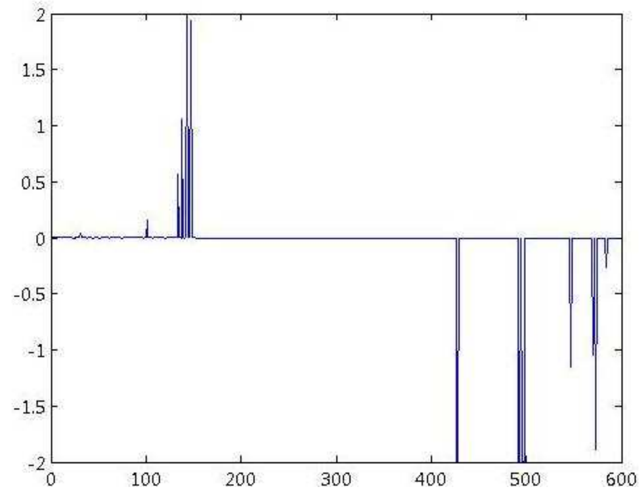

Fig. 11. Error in validation set with 596 samples

2 Conclusion

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

Having confronted with such problem we were tended to design a novel technique that takes advantages of Neural Networks in learning nonlinear relations and combines it with Kalman Filter Theory to be useful in such noisy conditions. On the other hand, not directly getting involved in image processing procedure makes this technique useful for systems where their input must be provided by other resources and sensors.

The results of implementation of mentioned technique in MRL Middle Size Robots is considerable and suggest that the combined approach performs significantly better than traditional techniques.

References

1. J. Ruiz-del-Solar, P.A. Vallejos, "Motion Detection and tracking for an AIBO robot using camera motion compensation and Kalman filtering," Robocup Symposium, Bremen 2006.
2. V. Spinko, D. Shi, and W. S. Ng, "Endoscope tracking using Wavelet-Gravitation Network incorporated with Kalman Filter," 18th IEEE International Conference on Tools with Artificial Intelligence, 2006, in press.
3. X. Zhang and A.A. Minai, "Detecting corresponding segments across images using synchronizable Pulse-Coupled neural networks," Proceedings IEEE/NNS International Joint Conference on Neural Networks, Washington, pp. 82C-825, 2001.
4. S. Ullman, "The interpretation of visual motion," MIT Press, London, 1979.
5. G. Lei, G. Baolong, "Visual system and distributed deduction theory," Xidan University Publisher, 1995.
6. B. Yu, L. Zhang, " Pulse Coupled Neural Network for Motion Detection," Proceedings of the International Joint Conference on. Vol. 2, pp. 1179 – 1184, July 2003.
7. G. Welch and G. Bishop, "An introduction to Kalman Filter," UNCChapel Hill, 5 April 2004.
8. Z.Neji, F.M.Beji, "Neural Network and time series identification and prediction," IEEE-INNS-ENNS International Joint Conference on Neural Networks, 2000.
9. P. J. Werbos, “Beyond regression: new tools for prediction and analysis in the behavioral sciences,” Ph.D. thesis, Harvard Univ., Cambridge, MA, 1974.