2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Visual State Estimation Using Self-Tuning Kalman Filter and Echo State Network

Chi-Yi Tsai, Xavier Dutoit, Kai-Tai Song, Hendrik Van Brussel and Marnix Nuttin

Abstract—**This paper presents a novel design of visual state estimation for an image-based tracking control system to estimate system state during visual tracking control process. The advantage of this design is that it can estimate the target status and target image velocity without using the knowledge of target's 3D motion-model information. This advantage is helpful for real-time visual tracking controller design. In order to increase the robustness against random observation noise, a neural network based self-tuning algorithm is proposed using echo state network (ESN) technique. The visual state estimator is designed by combining a Kalman filter with the ESN-based self-tuning algorithm. The performance of this estimator design has been evaluated using computer simulation. Several interesting experiments on a mobile robot validate the proposed algorithms.**

I. Introduction

VISION systems have been widely used as perception sensors for autonomous intelligent robots and the research on visual tracking control of a mobile robot to track a target of interest has been an active area of robotic research [1-3]. The visual tracking task of a mobile robot encompasses several key factors such as motion control, target detection, depth estimation, position and velocity estimations, etc. In [4], the authors suggested that the prediction of target motion can help the visual tracking system to track the target within the camera's field of view. However, in their design, the proposed estimator only can estimate the target motion in eight directions. In practical applications, a mobile robot usually needs to track a dynamic moving target. Therefore, a visual state estimator to estimate the motion of a dynamic moving target can greatly enhance the performance of a visual tracking control system.

In this paper, we address the problem regarding position and velocity estimation of a dynamic motion target in the image plane. In the existing estimation methods, it is well known that a Kalman filter is one of the best linear estimators for a linear plant model with Gaussian white noise [5]. However, if the noise statistics are unknown, it will be difficult to determine suitable covariance matrices for computing the Kalman gain matrix [6]. Thanks to the neural network techniques, the observation noise statistics can be estimated by an artificial neural network without any noise model [7]. Therefore, a neural network based self-tuning algorithm is helpful for a Kalman filter to work in an environment with unknown observation noise statistics.

There exist numerous neural network architectures. Amongst them, feedforward neural networks (FNNs) are the most popular models; however, FNNs only implement static input-output mappings. On the contrary, recurrent neural networks (RNNs) are better fit for time-dependent and non-reactive tasks, such as the one considered here, as the recurrent connections allow for some short-term memory. However, a major issue with RNNs is the training complexity. Recently, a new technique to use RNNs has been proposed: the Echo State Networks (ESNs) [8]. The idea of ESN is to use a large RNN while training only the readout layer. The recurrent part is created a priori and left fixed, and a simple linear memory-less readout is trained to project the state of the recurrent part onto the desired output. Thus the training complexity comes down to a one-step linear training, guaranteed to find a global optimum. This advantage motivates us to adopt ESN technique to filter the noise and estimate the noise variance.

In this paper, a novel visual state estimator is proposed by using ESN-based self-tuning Kalman filter technique. The ESN aims to filter the observation noise and provide the corresponding covariance matrix for the Kalman filter to estimate the optimal system state. Simulation and experimental results will be presented to validate the estimation performance as well as the robustness of proposed ESN-based self-tuning Kalman filter in visual tracking.

II. Problem Formulation

A. Visual Interaction Model

We first introduce the scenario under consideration. As shown in Fig 1, the considered system is a wheeled mobile robot equipped with a tilt camera mounted on top of it to track a moving target, such as a human face, in the image plane. The optical-axis of the camera faces the target of interest. Fig. 1 (a) illustrates the model of the mobile robot and target in the world coordinate frame F_f, in which the motion of the target is supposed to be holonomic with zero angular motion relative to the robot. Fig. 1 (b) is the side view of the scenario under consideration, in which the tilt angle ϕ gives the relationship between the camera coordinate frame F_c and the mobile coordinate frame F_m. In order for the mobile robot to interact

C.-Y. Tsai and K.-T. Song are with the Department of Electrical and Control Engineering, National Chiao Tung University, 1001 Ta Hsueh Road, Hsinchu 300, Taiwan R.O.C. (corresponding author: Kai-Tai Song; +886-3-5731865; fax: +886-3-5715998; e-mail: ktsong@mail.nctu.edu.tw).

X. Dutoit, H. Van Brussel and M. Nuttin are with the Department of Mechanical Engineering, Division PMA, K.U.Leuven, Celestijnenlaan 300B, B-3001 Leuven, Belgium. (e-mail: xavier.dutoit@mech.kuleuven.be).

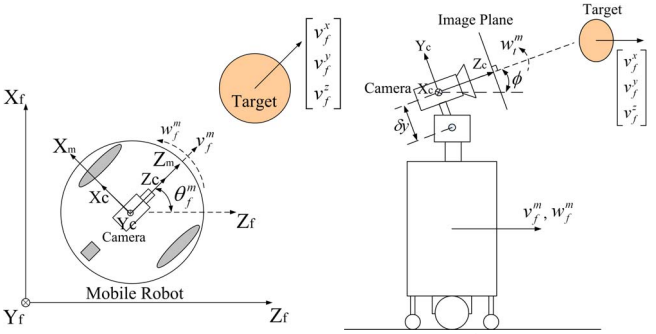

Fig. 1 (a) A model of the unicycle-modeled mobile robot and the target in world coordinate frame. (b) Side view of the mobile robot with a tilt camera mounted on top of it to track a dynamic target.

with the target in the image coordinate frame, a visual interaction model was proposed in authors' previous work [9]. Figure 2 shows the definition of observed system states in the image plane, which is used to derive the visual interaction model. In Fig. 2, x_i and y_i, respectively, are the horizontal and vertical position of the centroid of target in the image plane, and d_x is the width of target in the image plane. Let $X_i = [x_i \quad y_i \quad d_x]^T$ denote the system states in the image plane, (f_x, f_y) represent the fixed focal length along the image x-axis and y-axis, respectively, and W denotes the actual width of the target. The visual interaction model between robot and target in the image coordinate frame can be modeled as a dual-Jacobian equation such that [10]

$$\dot{X}_i = \dot{X}_i^t + \dot{X}_i^m = \mathbf{J}_t V_t + \mathbf{B}_i u, \tag{1}$$

where

$$\mathbf{J}_t = \begin{bmatrix} -k_x(\frac{x_i}{f_x}\cos\phi\sin\theta_f^m + \cos\theta_f^m) & -k_x\frac{x_i}{f_x}\sin\phi & -k_x(\frac{x_i}{f_x}\cos\phi\cos\theta_f^m - \sin\theta_f^m) \\ -k_y(\frac{y_i}{f_y}\cos\phi\sin\theta_f^m + \sin\phi\sin\theta_f^m) & -k_y(\frac{y_i}{f_y}\sin\phi - \cos\phi) & -k_y(\frac{y_i}{f_y}\cos\phi\cos\theta_f^m + \sin\phi\cos\theta_f^m) \\ -k_x\frac{d_x}{f_x}\cos\phi\sin\theta_f^m & -k_x\frac{d_x}{f_x}\sin\phi & -k_x\frac{d_x}{f_x}\cos\phi\cos\theta_f^m \end{bmatrix}$$

is termed *target image Jacobian* and transfers the target velocity $V_t = [v_f^x \quad v_f^y \quad v_f^z]^T$ into target image velocity $\dot{X}_i^t = [\dot{x}_i^t \quad \dot{y}_i^t \quad \dot{d}_x^t]^T = \mathbf{J}_t V_t$, and

$$\mathbf{B}_i = \begin{bmatrix} \frac{k_x}{f_x}x_i\cos\phi & (\frac{x_i^2+f_x^2}{f_x})\cos\phi - \frac{f_x}{f_y}(k_y\delta y + y_i)\sin\phi & -\frac{x_i(k_y\delta y + y_i)}{f_y} \\ k_y(\sin\phi + \frac{y_i}{f_y}\cos\phi) & \frac{f_y}{f_x}x_i(\sin\phi + \frac{y_i}{f_y}\cos\phi) & -\frac{y_i^2+f_y^2+k_y y_i\delta y}{f_y} \\ \frac{k_x}{f_x}d_x\cos\phi & \frac{x_i d_x}{f_x}\cos\phi & -\frac{d_x(k_y\delta y + y_i)}{f_y} \end{bmatrix}$$

is termed *robot image Jacobian* and transfers the mobile robot control velocity $u = [v_f^m \quad w_f^m \quad w_t^m]^T$ into robot image velocity $\dot{X}_i^m = \mathbf{B}_i u$; $k_x = d_x/\mathrm{W}$ and $k_y = k_x f_y / f_x$ are two scalars.

B. Visual Tracking Control

Based on the visual interaction model (1), a feedback control law can be found by feedback linearization such that

$$u = \mathbf{B}_i^{-1}(\mathbf{K}_g X_e - \mathbf{J}_t V_t) = \mathbf{B}_i^{-1}(\mathbf{K}_g X_e - \dot{X}_i^t), \tag{2}$$

where $X_e = [x_e \quad y_e \quad d_e]^T = [\bar{x}_i - x_i^* \quad \bar{y}_i - y_i^* \quad \bar{d}_x - d_x^*]^T$ is the error coordinates defined in the image plane, in which $\bar{X}_i^d = [\bar{x}_i \quad \bar{y}_i \quad \bar{d}_x]^T$ is the vector of fixed desired states in the image plane, and $X_i^* = [x_i^* \quad y_i^* \quad d_x^*]^T$ is the estimated state vector from a visual state estimator (see later). $\mathbf{K}_g = diag(\alpha_1, \alpha_2, \alpha_3) > 0$ is a 3-by-3 positive gain matrix. The visual tracking control law (2) indicates that the controller needs information about target status X_i and target image velocity $\dot{X}_i^t$. Therefore, a visual state estimator is required in order to obtain the optimal estimates of target status X_i and target image velocity $\dot{X}_i^t$ in the image space for visual tracking control task.

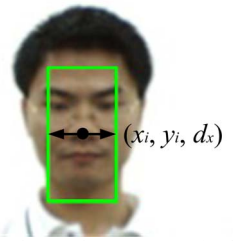

Fig. 2 Definition of the observed system state in the image plane.

C. The Visual State Estimation Problem

Because actual image processing is discrete, the first step of visual state estimator design is to discretize the system model (1) into the corresponding discrete form such that

$$X_i[n] = X_i[n-1] + \mathrm{T}\dot{X}_i^t[n-1] + \mathrm{T}\mathbf{B}_i u_{n-1}, \text{ for } n = 1, 2, \ldots \tag{3}$$

where T denotes the sampling time of the digital system, and u_n is the discrete-time control signal at time step n. Suppose that the target's motion can be approximated as a smooth motion during a sampling time, then the target image velocity has the following result

$$\dot{X}_i^t[n] = \dot{X}_i^t[n-1]. \tag{4}$$

Based on (3) and (4), the propagation model can be obtained such that

$$X_n = \begin{bmatrix} \mathbf{I}_3 & \mathrm{T}\mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} X_{n-1} + \begin{bmatrix} \mathrm{T}\mathbf{B}_i \\ \mathbf{0}_3 \end{bmatrix} u_{n-1} \equiv \mathbf{A}_{est} X_{n-1} + \mathbf{B}_{est} u_{n-1}, \tag{5}$$

where $X_n = [(X_i[n])^T \quad (\dot{X}_i^t[n])^T]^T$ is the vector of system estimates at time step n, $\mathbf{I}_3$ is a 3-by-3 identity matrix, and $\mathbf{0}_3$ is a 3-by-3 zero matrix. Next, since the observed image contains only information about target status X_i at each time step, the observation model is given by

$$Z_n = [\mathbf{I}_3 \quad \mathbf{0}_3] X_n \equiv \mathbf{H}_{est} X_n. \tag{6}$$

Based on (5) and (6), the visual state estimation problem is defined as to find the state estimate X_n^* that minimizes the weighted least square criterion:

$$X_n^* = \arg\min_X [(X_n - X)^T \mathbf{P}_n^{-1}(X_n - X) + (Z_n - \mathbf{H}_{est}X)^T \mathbf{R}_n^{-1}(Z_n - \mathbf{H}_{est}X)] \tag{7}$$

where $\mathbf{P}_n = \mathbf{A}_{est}\mathbf{P}_{n-1}\mathbf{A}_{est}^T$ is the covariance matrix of propagation model (5) at time step n, and $\mathbf{R}_n$ is the covariance matrix of observation model (6) at time step n.

III. Self-Tuning Kalman Filter

Define that (X_n,$\mathbf{P}_n$) are the propagation state and the

corresponding covariance matrix at time step n, $(X^*_{n-1}, \mathbf{P}^*_{n-1})$ are the optimal estimate and the corresponding covariance matrix at time step n-1, $\delta X_n = [(\delta X_i[n])^T \quad (\delta \dot{X}_i^t[n])^T]^T$ represents Gaussian propagation uncertainty with zero mean and covariance matrix $\mathbf{Q}_n$ at time step n, and δZ_n represents Gaussian observation uncertainty with zero mean and covariance matrix $\mathbf{R}_n$ at time step n. Then, when the linear propagation model (5) and the linear observation model (6) both have Gaussian propagation and observation uncertainties

$$\text{Propagation: } X_n = \mathbf{A}_{est} X^*_{n-1} + \mathbf{B}_{est} u_{n-1} + \delta X_{n-1}, \quad (8)$$

$$\text{Covariance Propagation: } \mathbf{P}_n = \mathbf{A}_{est} \mathbf{P}^*_{n-1} \mathbf{A}^T_{est} + \mathbf{Q}_{n-1}, \quad (9)$$

$$\text{Observation: } Z_n = \mathbf{H}_{est} X_n + \delta Z_n, \quad (10)$$

a Kalman filter will provide the local minimum solution of performance criterion (7) and the corresponding covariance matrix at time step n such that [5]

$$X^*_n = X^p_n + \mathbf{K}_n (Z_n - \mathbf{H}_{est} X^p_n) \text{ and } \mathbf{P}^*_n = (\mathbf{I}_6 - \mathbf{K}_n \mathbf{H}_{est}) \mathbf{P}_n, \quad (11)$$

where $X^p_n = \mathbf{A}_{est} X^*_{n-1} + \mathbf{B}_{est} u_{n-1}$ is the ideal propagation state, $\mathbf{K}_n = \mathbf{P}_n \mathbf{H}^T_{est} (\mathbf{H}_{est} \mathbf{P}_n \mathbf{H}^T_{est} + \mathbf{R}_n)^{-1}$ is the Kalman gain matrix, and $\mathbf{I}_6$ is a 6-by-6 identity matrix.

According to [6], the performance of a Kalman filter is determined by the covariance matrices $\mathbf{Q}_n$ and $\mathbf{R}_n$. Thus, a difficult problem in Kalman filter applications is how to determine the values of matrices $\mathbf{Q}_n$ and $\mathbf{R}_n$ for computing the Kalman gain matrix $\mathbf{K}_n$. Typically, this problem is left up to engineering intuition by a trial-and-error procedure. However, in robotic applications the observation uncertainty usually varies with the conditions of target motion (such as orientation and rotation of a tracked human face) and working environment (such as light variation and occlusion), and the corresponding covariance matrix $\mathbf{R}_n$ are time-varying for different operating conditions. In order to deal with this problem, the neural network techniques are useful to filter the observation noise and estimate the noise variance without any noise model [7]. Therefore, this advantage motivates us to combine a neural network based self-tuning algorithm with a Kalman filter to filter the observation noise and provide a suitable observation covariance matrix $\mathbf{R}_n$ in the varying environmental conditions. Figure 3 shows the block diagram of the proposed ESN-based self-tuning Kalman filter, in which $\hat{Z}_n$ denotes the measurement with observation noise, and (Z_n, $\Delta\mathbf{R}_n$) are the filtered measurement and the estimated noise covariance matrix. The covariance matrix of the observation signal is then updated such that

$$\mathbf{R}_n = \mathbf{R}_0 + \Delta\mathbf{R}_n, \quad (12)$$

where $\mathbf{R}_0$ is a fixed initial covariance matrix to avoid the covariance matrix becoming null. In the following section, we will present the design of ESN-based self-tuning algorithm.

Note that because we do not have an exact mathematic model to describe the propagation of the uncertainty, the propagation covariance matrix $\mathbf{Q}_n$ is supposed to be fixed without updating in this design.

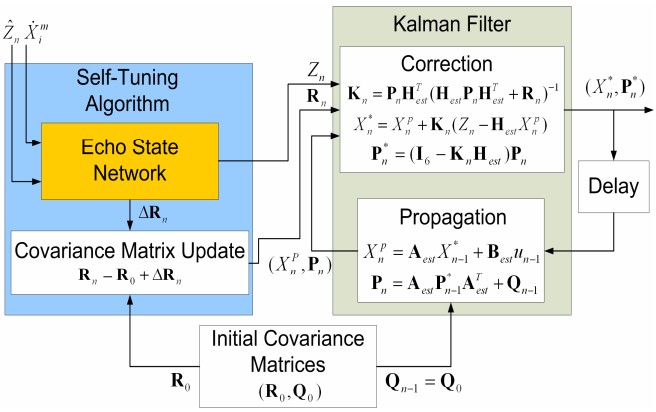

Fig. 3 Proposed neural network based self-tuning Kalman filter.

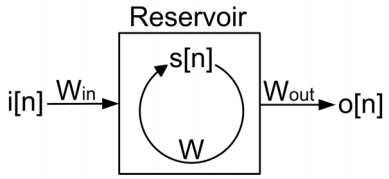

Fig. 4 General structure of an ESN, in which $s[n]$ is the state of every neuron in the reservoir, and $\mathbf{W}$ is the connection matrix between every neuron.

IV. Echo State Network for Noise Filtering and Noise Variance Estimation

We will now describe the neural network used in the current scenario. An ESN is described by an input matrix $\mathbf{W}_{IN}$, a connection matrix $\mathbf{W}$ and a linear readout $\mathbf{W}_{OUT}$ (see Fig. 4).

A. Activation Function

The ESN runs in discrete time. At each time step, the state vector (describing the activation level of every neuron) is updated according to

$$\mathbf{s}[n+1] = f(m \cdot (\mathbf{W}_{in} \cdot \mathbf{i}[n] + \mathbf{W} \cdot \mathbf{s}[n]) + (1-m) \cdot \mathbf{s}[n]), \quad \forall n > 0 \quad (13)$$

where $\mathbf{i}[n]$ is the current input vector, $\mathbf{s}[n]$ is the current state (with $\mathbf{s}[0]$=0), $f(.)$ is a non-linear function (here we use a hyperbolic tangent) and m is a parameter controlling the *leaking rate* of each neuron. At each time step, the output is given by

$$o[n] = \mathbf{W}_{out} \cdot \begin{bmatrix} \mathbf{s}[n] \\ 1 \end{bmatrix} \quad (14)$$

B. Network Creation

The matrices $\mathbf{W}_{in}$ and $\mathbf{W}$ are created randomly. The connection from the inputs should have weights large enough to have sufficient effect inside the reservoir and small enough not to drive the reservoir to saturation [11]. An efficient trade-off has been found by setting the elements of $\mathbf{W}_{in}$ to -0.1 or +0.1 with equal probability. The reservoir connections must guarantee the echo state property [8]. Intuitively, this property states that the initial conditions have an asymptotically decreasing influence on the current state of the network. To do so, the elements of $\mathbf{W}$ are drawn from a normal distribution, and the whole matrix is then re-scaled to make its spectral radius equal to 0.9.

C. Training

The output matrix $\mathbf{W}_{\text{out}}$ is created then during the training. As the output at each time step is given by (14), the training is done by solving

$$\mathbf{W}_{\text{out}} \cdot \begin{bmatrix} \mathbf{s}[1] & \mathbf{s}[2] & \cdots & \mathbf{s}[n_t] \\ 1 & 1 & \cdots & 1 \end{bmatrix} = [\hat{o}[1] \quad \hat{o}[2] \quad \cdots \quad \hat{o}[n_t]] \quad (15)$$

in the mean square sense (n_t being the number of time samples and $\hat{o}[n]$ the desired output at time step n).

D. ESN-based Self-Tuning Algorithm

In the current implementation, we use 3 independent ESNs, one for each parameter x_i, y_i and d_x. Each ESN receives as input the corresponding measurement with noise $\hat{Z}_n$ and the corresponding robot image velocity $\dot{X}_i^m$. It is then trained to output at each time step an estimate of the actual measurement Z_n (see Fig. 5).

To estimate the variance of the noise at time step n, we take in the present design the variance of the time series (recorded over time with length N) of observation noise $\delta Z_n = \hat{Z}_n - Z_n$. Let δZ_x, δZ_y and δZ_d denote the time series of observation noise corresponding to x_i, y_i and d_x, the covariance matrix of observation noise at time step n is estimated by

$$\Delta\mathbf{R}_n = \text{diag}(\text{var}(\delta Z_x), \text{var}(\delta Z_y), \text{var}(\delta Z_d)), \quad (16)$$

where var(x) denotes the variance value of vector x. In the current design, the time series length (N) is set to 9. The cross-covariance values of Z_n are supposed to be zero since three independent ESNs are used.

V. Simulation and Experimental Results

A. Simulation Setup

In order to evaluate the performance of the proposed visual state estimator, a simulation environment has been setup using MATLAB. Figure 6 shows the architecture of the simulation setup. In Fig. 6, X_n denotes the reference signal needed to be estimated by a visual state estimator. The input of the visual state estimator is the observation signal $\hat{Z}_n$ with random noise (RN)

$$RN = \begin{cases} K_n\sigma_1(0.5-\sigma_2), & if \ (\sigma_3 < \rho) \\ (1+\sigma_1)(0.5-\sigma_2), & otherwise \end{cases} \quad (17)$$

where $K_n > 1$ is the noise gain; $\sigma_i \in [0,1]$, i=1~3, are three random signals with uniform distribution; and $\rho \in [0,1]$ is a constant threshold value. Expression (17) indicates that the intensity of RN is time-varying and dependent on a random condition. If the condition $(\sigma_3 < \rho)$ is satisfied, then RN will have large noise gain; otherwise RN will only have noise gain smaller than 2. Thus, the threshold value ρ determines the probability of the event of appearing large observation noise. For example, if $\rho = 1$, then the observation signal will always have the largest noise intensity. This kind of noise usually happens during the practical visual tracking process of the mobile robot, since the intensity of observation uncertainty usually is position-dependent and light-dependent.

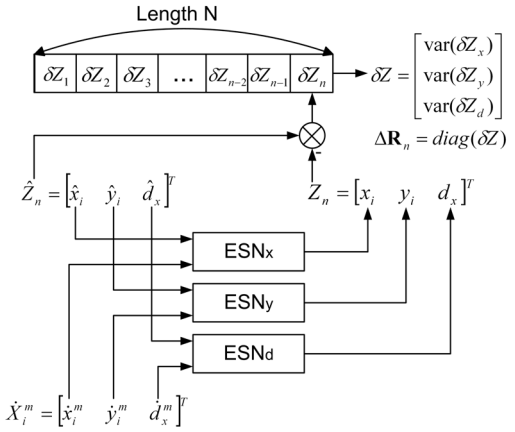

Fig. 5 Inputs and outputs of the ESNs (detail of the ESN box from Fig. 3)

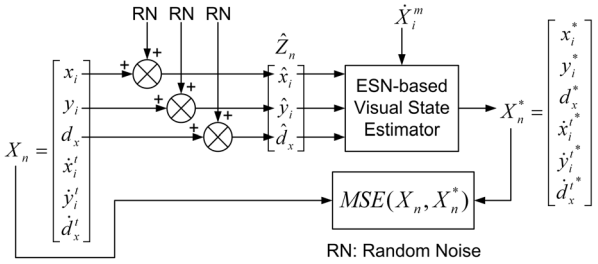

Fig. 6 Simulation setup for the performance evaluation of the visual state estimator shown in Fig. 3 (see Section III for the details).

In the following, a visual state estimator is utilized to filter RN and provide the optimal estimation. The performance of the visual state estimator is then validated by mean-squared-error (MSE) criterion between the ideal signal X_n and the estimated signal X_n^*. Table I shows the parameters used in the simulations. Note that we use a threshold $\rho = 0.75$ when generating the training data for the ESNs. Moreover, the parameters used to create the ESNs (found empirically) are n_r=90 neurons (for all three ESNs) and m=0.5, 0.6 and 0.8 for x_i, y_i and d_x respectively.

B. Simulation Results

Three visual state estimators are used to compare the performance: Kalman filter (KF), self-tuning Kalman filter using linear regression (STKF-LR) [10], and the proposed self-tuning Kalman filter using ESN (STKF-ESN). Table II shows the average results of MSE measurements as the threshold value $\rho = 1$ and $\rho = 0$ in the simulations (out of 40 simulations for each ρ). In Table II, the bold font denotes the smallest value of the MSE measurement across each row. From Table II, we observe that the estimation results of KF and STKF-LR are very sensitive to the intensity of the observation noise. As the threshold value ρ increased from 0 to 1, the average MSE measurements are also increased significantly. Moreover, when the threshold value $\rho = 1$ (the observation signal always has the largest noise intensity), the proposed STKF-ESN provides the best estimation results

TABLE I
PARAMETERS USED IN THE SIMULATIONS AND EXPERIMENTS

Symbol	Quantity	Description
(f_x, f_y)	(393.4,391.8) pixels	Camera focal length in retinal coordinates.
W	12 cm	Width of the target.
D	40 cm	Distance between two drive wheels.
T	80 ms	Sampling period of the control system.
δy	10 cm	Distance between the robot head and the camera
$(\bar{x}_i, \bar{y}_i, \bar{d}_x)$	(0,0,35)	Desired system state in image plane.
$(\alpha_1, \alpha_2, \alpha_3)$	(1,3/2,2/5)	Three distinct positive constants.
$\mathbf{Q}_0$	diag(5,5,5,20,20,20)	Initial propagation covariance matrix
$\mathbf{R}_0$	diag(5,5,5)	Initial observation covariance matrix
K_n	10	Noise gain

TABLE II
AVERAGE MSE MEASUREMENTS OF COMPUTER SIMULATIONS

	MSE Value	KF	STKF-LR	STKF-ESN
x_i	$\rho = 1$	1.1979	1.8969	**0.7885**
	$\rho = 0$	0.1766	0.6639	**0.1720**
	MSE Gap	1.0212	1.2330	**0.6164**
y_i	$\rho = 1$	1.1488	1.3223	**0.5644**
	$\rho = 0$	**0.1544**	0.3160	0.3076
	MSE Gap	0.9944	1.0063	**0.2568**
d_x	$\rho = 1$	4.4150	2.7493	**0.9879**
	$\rho = 0$	0.1825	0.1951	**0.1404**
	MSE Gap	4.2324	2.5542	**0.8476**
$\dot{x}_i^t$	$\rho = 1$	18.0588	23.5390	**16.0603**
	$\rho = 0$	13.6167	17.2235	**13.4824**
	MSE Gap	4.4421	6.3155	**2.5778**
$\dot{y}_i^t$	$\rho = 1$	6.1635	5.2398	**2.2938**
	$\rho = 0$	**1.4735**	2.0022	1.6126
	MSE Gap	4.6900	3.2376	**0.6812**
$\dot{d}_x^t$	$\rho = 1$	14.9345	6.1500	**1.5352**
	$\rho = 0$	0.6867	0.7696	**0.4380**
	MSE Gap	14.2479	5.3805	**1.0972**

compared with the other two estimators. Note that STKF-LR uses the measurement offset for the computation of observation variance. Please refer to [10] for more details.

Table II also records the MSE gap between $\rho = 1$ and $\rho = 0$. A small MSE gap implies a large robustness against the intensity of observation noise. Table II shows the MSE gaps of KF and STKF-LR for all estimates are larger than that of STKF-ESN. This implies that the proposed STKF-ESN provides high robustness against the observation uncertainty compared with KF and STKF-LR. Therefore, the simulation results validate the performance and robustness of the proposed ESN-based visual state estimator.

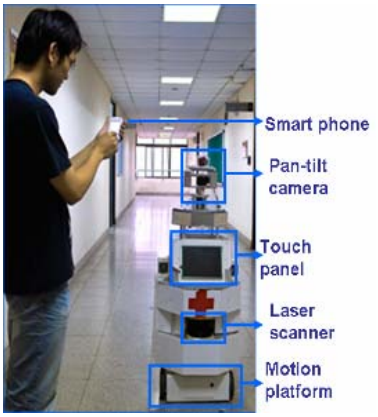

Fig. 7 An elder-care mobile robot, *Rola*, used in the experiments.

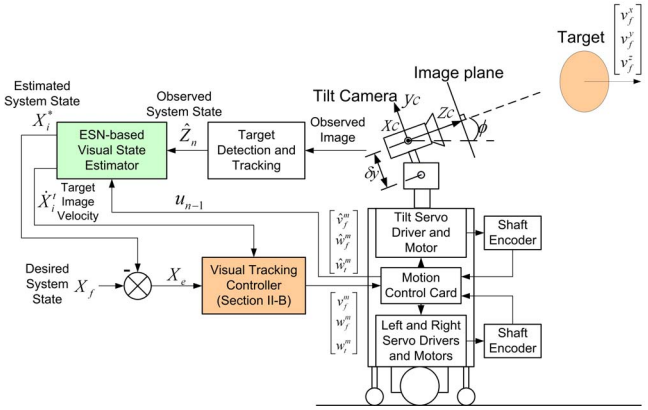

Fig. 8 Block diagram of the implemented visual tracking control system, which includes the proposed ESN-based visual state estimator.

C. *Experiments*

Figure 7 shows the experimental mobile robot, *Rola*, used in the experiments. *Rola* stands for *robot of living aid* designed to provide an elder immediate medical care. It includes several functions such as location-aware detection, pose estimation, visual tracking and video transmission. For visual tracking and video transmission functions, a pan-tilt USB camera is mounted on the robot to detect and track user's face. In the experiments, the linear and angular command velocities (v_f^m, w_f^m) are used to control the motion of the mobile robot and the tilt command velocity w_t^m is used to control the tilt angle of the pan-tilt camera. Figure 8 depicts the implemented visual tracking control system utilizing the proposed ESN-based visual state estimator to estimate the system state and target image velocity. The processing time of the visual tracking system is less than 80ms including image processing, estimator and controller computation. Thus, the overall tracking system can track user's face in real-time.

D. *Experimental Results*

In the experiments, *Rola* aims to track the user's face in a practical environment with hand-controlled light-variation situations, which make the intensity of observation noise position-dependent. Thus, the proposed ESN-based visual state estimator plays an important role in overcoming the position-dependent observation noise. Note that the ESN parameters used in the experiments are the same as that used

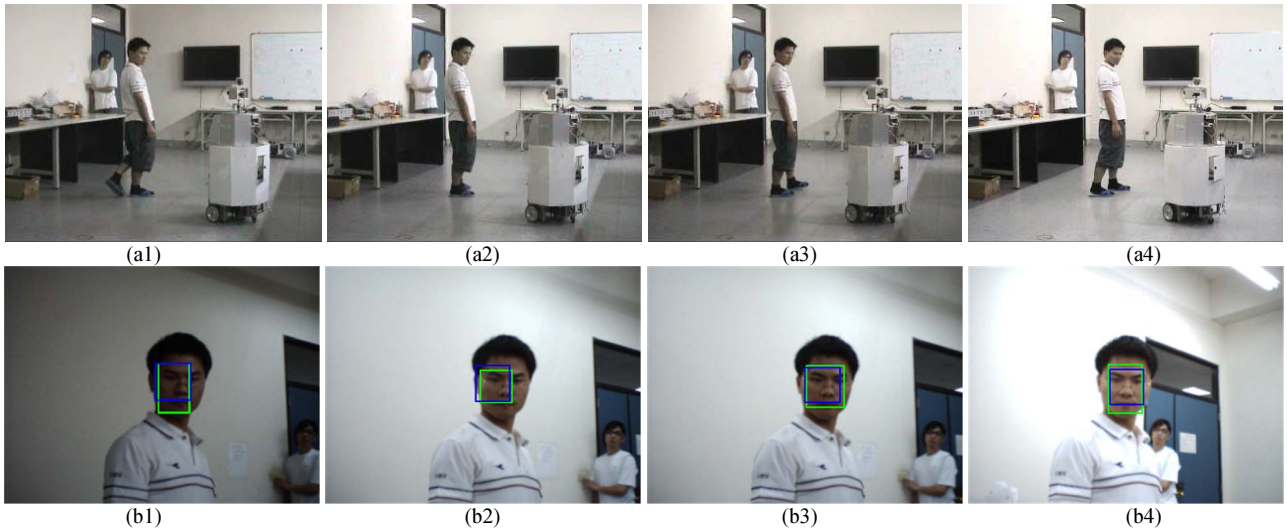

Fig. 9. Experimental results. (a1-a4): Image sequence recorded from a DV camera. (b1-b4): Corresponding image sequence recorded from the on-board USB camera. In the pictures (b1-b4), the green window indicates the observation, and the blue window is the corresponding output of ESNs.

in the simulations.

Figure 9 shows the experimental results of the implemented visual tracking control system given in Fig. 8. Figures 9(a1-a4) illustrate recorded pictures from a digital video (DV) camera, and Figs. 9(b1-b4) show the corresponding pictures recorded by the on-board USB camera. In Figs. 9(a1-a4), the tracked person was walking in an environment with light-variation, and the robot tracked the person's face as expected. As shown in Figs. 9(b1-b4), the person's face suddenly became lighter due to the variation in illumination. In such situations, the ESNs work to provide a stable output even when the observation contains rapid random noises. Therefore, the robot estimated and tracked the person's face in the image plane successfully. Note that the image sequence shown in Fig. 9 is about 2 seconds. Several video clips of mobile robot visual tracking experimental results are available online in [12].

VI. Conclusion

In this paper, a novel visual state estimator is proposed based on ESN-based self-tuning Kalman filter technique. This design can be applied to several visual tracking applications, such as visual tracking control, visual surveillance, and visual navigation, etc., to estimate the position and the velocity of the target in the image plane. Simulations show that this design provides high robustness against the observation uncertainty with time-varying intensity. This advantage is very useful in robotic applications, since the observation uncertainty usually varies with the conditions of target motion and working environment. Computer simulation results validate the robustness and performance of proposed estimation method by comparing with conventional Kalman filter and linear regression based self-tuning Kalman filter. Moreover, experimental results also verify the tracking performance of the proposed visual tracking system in a practical environment under light-varying conditions.

Acknowledgment

This work was supported by National Science Council of Taiwan, ROC under grant NSC 95-2218-E-009-008, 95-EC-17-A-04-S1-054; and the research foundation Flanders FWO, Belgium under grant G.0317.05.

References

[1] G. L. Mariottini, G. Oriolo, and D. Prattichizzo, "Image-based visual servoing for nonholonomic mobile robots using epipolar geometry," *IEEE Trans. on Robotics*, Vol. 23, No. 1, pp. 87-100, 2007.

[2] J. Chen, W. E. Dixon, D. M. Dawson and M. McIntyre, "Homography-based visual servo tracking control of a wheeled mobile robot," *IEEE Trans. on Robotics*, Vol. 22, No. 2, pp. 407-416, 2006.

[3] T. Nierobisch, W. Fischer, and F. Hoffmann, "Large view visual servoing of a mobile robot with a pan-tilt camera," in *Proc. IEEE/RSJ Int. Conf. on Intel. Rob. and Sys.*, Beijing, China, pp. 3307-3312, 2006.

[4] L.-Q. Xu and D. C. Hogg, "Neural networks in human motion tracking – An experimental study," *Journal of Image and Vision Computing*, Vol. 15, No. 8, pp. 607-615, 1997.

[5] J. D. Schutter, J. D. Geeter, T. Lefebvre and H. Bruyninckx, "Kalman filters: a tutorial," *Journal A*, Vol. 40, No. 4, pp. 52-59, 1999.

[6] O. V. Korniyenko, M. S. Sharawi and D. N. Aloi, "Neural network based approach for tuning Kalman filter," in *Proc. IEEE Int. Conf. on Electro/Information Technology*, Lincoln, Nebraska, pp. 1-5, 2005.

[7] S.-S. Xiong and Z.-Y. Zhou, "Neural filtering of colored noise based on Kalman filter structure," *IEEE Trans. on Instrumentation and Measurement*, Vol. 52, No. 3, pp. 742-747, 2003.

[8] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks," German National Research Center for Information Technology, Tech. Rep., 2001.

[9] C.-Y. Tsai and K.-T. Song, "Face tracking interaction control of a nonholonomic mobile robot," in *Proc. IEEE/RSJ Int. Conf. on Intel. Rob. and Sys.*, Beijing, China, pp. 3319-3324, 2006.

[10] C.-Y. Tsai, K.-T. Song, X. Dutoit, H. Van Brussel and M. Nuttin, "Robust mobile robot visual tracking control system using self-tuning Kalman filter," in *Proc. IEEE Int. Sym. on Comp. Intel. in Rob. and Auto.*, Jacksonville, Florida, pp 161-166, 2007.

[11] H. Jaeger, "A tutorial on training recurrent neural networks, covering bppt, rtrl, ekf and the "echo state network" approach," German National Research Center for Information Technology, Tech. Rep., 2002.

[12] The experiment video website. [Online]. Available: http://isci.cn.nctu.edu.tw/video/RVTS_ESN/