Asian Journal of Control, Vol. 12, No. 4, pp. 488–509, July 2010
Published online 18 May 2010 in Wiley InterScience (www.interscience.wiley.com) DOI: 10.1002/asjc.204

ROBUST FACE TRACKING CONTROL OF A MOBILE ROBOT 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 face tracking algorithm and visual state estimation for a mobile robot face tracking interaction control system. The advantage of this design is that it can track a user's face under several external uncertainties and estimate the system state without the knowledge about target's 3D motion-model information. This feature is helpful for the development of a real-time visual tracking control system. In order to overcome the change in skin color due to light variation, a real-time face tracking algorithm is proposed based on an adaptive skin color search method. Moreover, in order to increase the robustness against colored observation noise, a new visual state estimator is designed by combining a Kalman filter with an echo state network-based self-tuning algorithm. The performance of this estimator design has been evaluated using computer simulation. Several experiments on a mobile robot validate the proposed control system.

Key Words: Visual tracking control, visual state estimation, echo state network, face tracking, illumination variation.

I. INTRODUCTION

One of the questions of autonomous mobile robots is how the robot can interact with people naturally. In other words, the way for a mobile robot to interact with people will be an essential factor for the mboxapplications of a home/service robotic system. In recent years, vision systems have been widely adopted as perception sensors for autonomous intelligent robots. Among various applications of vision systems, visual tracking plays an important role in autonomous robot navigation and control. Thus the research on visual tracking control of a mobile robot to track a target of interest has been an active area of robotic research [1–6].

The purpose of this study is to develop a robust face tracking interaction control system using visual tracking control techniques for a wheeled mobile robot in human-robot interaction scenarios. The visual tracking task of a mobile robot encompasses several key factors such as target detection, robot motion control, depth estimation, image Jacobian estimation, target state estimation, etc. We divide these key factors into three fields: target detection, visual tracking control and visual state estimation. In target detection, in order forthe mobile robot to interact with people, an important factor is human identification and recognition, in which face analysis is one of the most interested research areas in computer vision [7]. This paper will

Manuscript received February 29, 2008; revised January 10, 2009; accepted March 10, 2009.

Chi-Yi Tsai is with the Department of Electrical Engineering, Tamkang University, 151 Ying-chuan Road, Tamsui, Taipei County 25137, Taiwan (e-mail: chiyi.ece91g@nctu.edu.tw; chiyi_tsai@mail.tku.edu.tw).

Kai-Tai Song is with the Department of Electrical Engineering, National Chiao Tung University, 1001 Ta Hsueh Road, Hsinchu 300, Taiwan.

Xavier Dutoit, Hendrik Van Brussel, and Marnix 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).

This work was supported by the National Science Council of Taiwan, 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.

focus on the design of face tracking which is an important issue in face analysis research. Traditional face tracking techniques need several cues such as shape, color, features and motion for processing. Nowadays, color-based object tracking has gained much attention because color is a sufficient cue to segment the face from background imagery [8–12]. However, color segmentation suffers from color variation caused by irregular illumination as well as the view of the camera.

In visual tracking control, the reported methods can be categorized into two groups based on motion constraints: the visual servoing for holonomic manipulators and the visual tracking for nonholonomic mobile robots. Visual servoing technique for holonomic manipulators has been investigated extensively and many powerful tools can be found in the literature [13, 14]. However, the results used in holonomic manipulators are not suitable for most mobile robots due to the nonholonomic motion constraints on the mobile platform. On one hand, many efforts have been made to deal with the design of mobile robot visual tracking controllers to track a static object [1–3, 15–17]. On the other hand, only a few efforts focus on the visual tracking control design of tracking a moving (non-static) target [4–6]. Notably, in order to enhance the tracking performance in the case of tracking a moving target, Xu and Hogg suggested that the prediction of target motion can help the visual tracking system to track the target within the camera's field of view [18]. Therefore, one of the challenges in visual tracking of a moving target is how to predict or estimate the motion of the moving target during visual tracking task execution. This problem motivates us to design a visual state estimator to provide the knowledge about target motion for enhancing the performance of a visual tracking control system.

The visual state estimation problem is well defined in machine vision, which aims to determine the position and velocity of an object moving in the 3D space by observing its motion in the 2D image plane of a perspective vision system (or pinhole camera). Traditionally, the existing methods usually suppose that the pinhole camera is fixed and the target is moving under Riccati dynamics in 3D space [19–23]. However, these assumptions cannot be satisfied in the issue addressed in this paper, since the camera and target both are moving with different dynamics. Recently, a dual-Jacobian visual interaction model has been proposed in [24, 25]. Based on this visual interaction model, the target image velocities can be used instead of the 3D motion velocities, and the visual state estimation problem can be resolved in the 2D image plane directly. In [24], the authors proposed two visual state estimators: one was developed under the condition of knowing the target 3D velocity, and the other was designed by releasing this condition. In [25], a self-tuning Kalman filter algorithm is proposed to estimate the target state and target image velocity without the knowledge about its motion in 3D space. It is well known that Kalman filter is one of the best linear estimators for a linear plant model with Gaussian white noise [26]; however, if the noise statistics are unknown, it will be difficult to determine suitable covariance matrices for computing the Kalman gain matrix [27]. Thanks to the neural network techniques, the observation noise statistics can be estimated by an artificial neural network without the knowledge of noise statistics [28]. 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) [29]. 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 the global optimum for a given ESN. 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 the 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. In the case of colored observation noise, traditional approaches need to extend the dimension of the original Kalman filter [30] or adjust the original observation equation [28] by a colored noise shaping filter. In contrast, the colored observation noise can be filtered into Gaussian white noise through the proposed ESN noise filter; thus, the robustness of the original Kalman filter can be improved without dimension extension or observation equation adjustment. Moreover, in order to overcome the irregular illumination

problem during the visual tracking process, a real-time face tracking algorithm based on a novel adaptive skin color searching method is proposed for face tracking under illumination variation. Therefore, a robust face tracking interaction control system can be achieved by combining the proposed real-time face tracking algorithm with the proposed ESN-based self-tuning Kalman filter. Simulation and experimental results will be presented to validate the estimation performance as well as the robustness of the proposed mobile robot face tracking interaction control system.

The rest of this paper is organized as follows. Section II describes the problem formulation and controller design for a wheeled mobile robot visual tracking control system. In Section III, the proposed ESN-based self-tuning Kalman filter and the design of ESN for noise filtering and variance estimation are presented. Section IV presents the proposed real-time face tracking algorithm under illumination variation. Simulation and experimental results are reported in Section V. Several simulations and experimental observations will be presented and discussed. Section VI concludes the contributions of this paper.

II. PROBLEM FORMULATION AND CONTROLLER DESIGN

Fig. 1 shows the visual tracking problem considered in this paper. In Fig. 1, a wheeled mobile robot equipped with a tilt camera on top of it aims 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 unicycle-modeled mobile robot and the 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.

2.1 Dual-Jacobian visual interaction model

In order for the mobile robot to interact with the target in the image coordinate frame, a dual-Jacobian visual interaction model was proposed in the authors' previous work [25]. Fig. 2 shows the definition of observed system states in the image plane used for the visual interaction model. In Fig. 2, x_i and y_i are the horizontal and vertical position of the centroid of target in the image plane, respectively, and d_x is the width of target in the image plane. Let $X_i = [x_i \;\; y_i \;\; d_x]^T$ denote the system states in the image plane, and $(f_x, \; f_y)$ represent fixed focal length along the image x-axis and y-axis, respectively, The visual interaction between robot and target in the image plane can be modeled as a dual-Jacobian equation such that

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

where

$$\mathbf{J}_i = \begin{bmatrix} -k_x\left(\frac{x_i}{\mathrm{f}_x}\cos\phi\sin\theta_f^m + \cos\theta_f^m\right) & -k_x\frac{x_i}{\mathrm{f}_x}\sin\phi & -k_x\left(\frac{x_i}{\mathrm{f}_x}\cos\phi\cos\theta_f^m - \sin\theta_f^m\right) \\ -k_y\left(\frac{y_i}{\mathrm{f}_y}\cos\phi\sin\theta_f^m + \sin\phi\sin\theta_f^m\right) & -k_y\left(\frac{y_i}{\mathrm{f}_y}\sin\phi - \cos\phi\right) & -k_y\left(\frac{y_i}{\mathrm{f}_y}\cos\phi\cos\theta_f^m + \sin\phi\cos\theta_f^m\right) \\ -k_x\frac{d_x}{\mathrm{f}_x}\cos\phi\sin\theta_f^m & -k_x\frac{d_x}{\mathrm{f}_x}\sin\phi & -k_x\frac{d_x}{\mathrm{f}_x}\cos\phi\cos\theta_f^m \end{bmatrix},$$

$$\mathbf{B}_i = \begin{bmatrix} \frac{k_x}{\mathrm{f}_x}x_i\cos\phi & \left(\frac{x_i^2+\mathrm{f}_x^2}{\mathrm{f}_x}\right)\cos\phi - \frac{\mathrm{f}_x}{\mathrm{f}_y}(k_y\delta y + y_i)\sin\phi & -\frac{x_i(k_y\delta y + y_i)}{\mathrm{f}_y} \\ k_y\left(\sin\phi + \frac{y_i}{\mathrm{f}_y}\cos\phi\right) & \frac{\mathrm{f}_y}{\mathrm{f}_x}x_i\left(\sin\phi + \frac{y_i}{\mathrm{f}_y}\cos\phi\right) & -\frac{y_i^2+\mathrm{f}_y^2+k_y y_i\delta y}{\mathrm{f}_y} \\ \frac{k_x}{\mathrm{f}_x}d_x\cos\phi & \frac{x_i d_x}{\mathrm{f}_x}\cos\phi & -\frac{d_x(k_y\delta y + y_i)}{\mathrm{f}_y} \end{bmatrix},$$

$k_x = d_x/\mathrm{W}$ and $k_y = k_x\mathrm{f}_y/\mathrm{f}_x$ are two scalars, W is the actual width of the target, $u = [v_f^m \;\; w_f^m \;\; w_t^m]^T$ is the vector of control velocities for the mobile

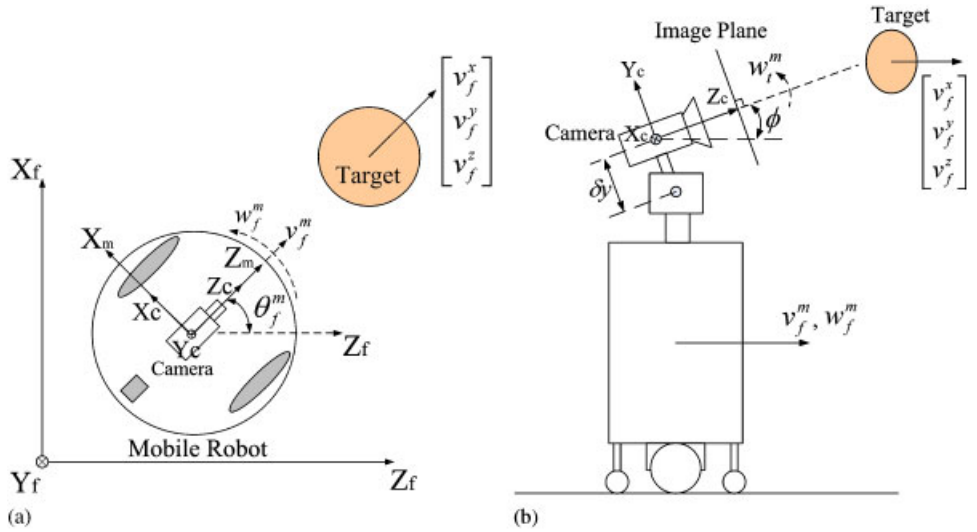

Fig. 1. (a) A model of the wheeled mobile robot and the target in the world coordinate frame and (b) Side view of the mobile robot with a tilt camera mounted on top of it to track a dynamic target.

robot and the tilt camera, and $V_f^t = [v_f^x \;\; v_f^y \;\; v_f^z]^T$ is the vector of target velocity in Cartesian coordinates. Expression (1) shows that the visual interaction model consists of two parts: the part of target motion $\dot{X}_i^t = [\dot{x}_i^t \;\; \dot{y}_i^t \;\; \dot{d}_x^t]^T = \mathbf{J}_i V_f^t$, and the effect of mobile robot motion $\dot{X}_i^m = [\dot{x}_i^m \;\; \dot{y}_i^m \;\; \dot{d}_x^m]^T = \mathbf{B}_i u$. Thus, matrix $\mathbf{J}_i$ is termed as 'target image Jacobian' transforming the target velocity V_f^t into target image velocity $\dot{X}_i^t$, and matrix $\mathbf{B}_i$ is termed as 'robot image Jacobian' transforming the mobile robot control velocity u into robot image velocity $\dot{X}_i^m$. In other words, the image velocity $\dot{X}_i$ is caused by a combination of target image velocity $\dot{X}_i^t$ and robot image velocity $\dot{X}_i^m$. Therefore, the visual interaction between robot and target in image coordinate frame can be modeled as a 'dual-Jacobian' visual interaction model (1).

2.2 Visual tracking control design

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

$$u = \mathbf{B}_i^{-1}(\mathbf{K}_g X_e - \mathbf{J}_i V_f^t), \tag{2a}$$

$$= \mathbf{B}_i^{-1}(\mathbf{K}_g X_e - \dot{X}_i^t). \tag{2b}$$

where $X_e = [x_e \;\; y_e \;\; d_e]^T = [\bar{x}_i - x_i^* \;\; \bar{y}_i - y_i^* \;\; \bar{d}_x - d_x^*]^T$ is the error coordinates defined in the image plane, in which $\bar{X}_i = [\bar{x}_i \;\; \bar{y}_i \;\; \bar{d}_x]^T$ is the vector of fixed desired states in the image plane, and $X_i^* = [x_i^* \;\; y_i^* \;\; d_x^*]^T$ is the vector of estimated states from a visual estimator (see later). $\mathbf{K}_g = \mathrm{diag}(\alpha_1, \alpha_2, \alpha_3) > 0$ is a 3-by-3 positive gain matrix, where $\mathrm{diag}(a, b, c)$ denotes a 3-by-3 diagonal matrix with diagonal element a, b, and c.

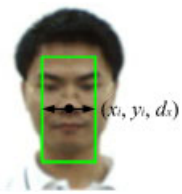

Fig. 2. Definition of the observed system state in image plane: x_i and y_i are the horizontal and vertical position of the centroid of target, respectively, and d_x is the width of target.

The visual tracking control law (2) indicates that the controller requires information about target 3D velocity V_f^t or target image velocity $\dot{X}_i^t$. If V_f^t is known, the first visual tracking control law (2a) only needs an estimate of target status X_i to calculate the control signal u. However, in practical applications, it is difficult to estimate V_f^t on-line in real time when using only one camera. In this situation, the second visual tracking control law (2b) provides a useful solution which only needs the target image velocity $\dot{X}_i^t$ in the image plane. Since the estimation of target 3D velocity is not considered in the current design, only the estimation of both target status X_i and target image velocity in the image space is realized for the later implementation of the visual tracking controller. This advantage will facilitate more general applications of the proposed tracking control scheme in the image plane. Note that the proposed visual tracking controller (2) can possess some degree of robustness against

system model uncertainties. For the discussion on the robustness analysis of the proposed controller, please refer to [31] for more technical details.

2.3 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

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

where T denotes the sampling time of the discrete system, and $u_n = [v_f^m \;\; w_f^m \;\; w_t^m]^T$ 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 period, 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 $X_n = [(X_i[n])^T \;\; (\dot{X}_i^t[n])^T]^T$ 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

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

Based on (5) and (6), the visual state estimation problem is defined as finding 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 USING ECHO STATE NETWORK

This section presents the design of the proposed ESN-based self-tuning Kalman filter to resolve the visual state estimation problem based on the performance criterion (7) described in Section 2.3.

3.1 ESN-based 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 $n-1$, $\delta X_n = [(\delta X_i[n])^T \;\; (\delta \dot{X}_i^t[n])^T]^T$ 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

Propagation: X_n

$$= \mathbf{A}_{est} X_{n-1}^* + \mathbf{B}_{est} u_{n-1} + \delta X_{n-1}, \tag{8}$$

Covariance Propagation: $\mathbf{P}_n$

$$= \mathbf{A}_{est} \mathbf{P}_{n-1}^* \mathbf{A}_{est}^T + \mathbf{Q}_{n-1}, \tag{9}$$

Observation: Z_n

$$= \mathbf{H}_{est} X_n + \delta Z_n, \tag{10}$$

a Kalman filter will provide the local minimum solution of performance criterion (7) and the corresponding covariance matrix at time step n such that [26]

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

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

According to [27], the filter 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 method. However, 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

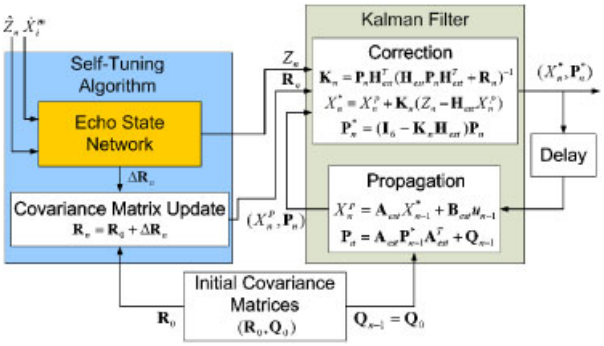

Fig. 3. Proposed ESN-based self-tuning Kalman filter.

different operating conditions. In order to deal with this problem, neural network techniques are useful to filter the observation noise and estimate the noise variance without the knowledge of noise statistics [28]. 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 varying environmental conditions. In this paper, ESN technique is adopted into the design of self-tuning algorithm due to the advantages described in Section I. Fig. 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

$$\mathbf{R}_n = \mathbf{R}_0 + \Delta\mathbf{R}_n, \tag{12}$$

where $\mathbf{R}_0$ is a fixed initial covariance matrix to avoid the covariance matrix becoming zeros. 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.

Remark 1. The observation noise usually is supposed to be Gaussian white noise in the Kalman filter design; however, the observation noise may be colored rather than white in most practical applications. In this case, the traditional method requires extending the dimensions of the original Kalman filter [30] or adjusting the original observation equation [28] by using a shaping filter in order to transform the colored noise into white noise. But this approach also needs the knowledge of colored noise statistics *a priori*. On the contrary, the proposed method employs ESN technique to filter the colored noise directly without knowledge of colored noise statistics. If the reservoir of the ESN contains sufficiently many neurons and those neurons are sufficiently decorrelated, due to the central limit theorem, the output noise correlation will decrease and the output noise will then tend to be Gaussian white noise. Since these two conditions are usually met in practice, the output of ESN can be directly used by Kalman filter without dimension extension or observation equation adjustment. Thus, the robustness of the original Kalman filter can be improved against not only Gaussian white noise, but also colored noise without any modification. This robustness property of the proposed ESN-based self-tuning Kalman filter will be validated in Section 5.1 and Section 5.3.

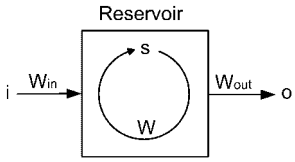

Fig. 4. General architecture of an ESN.

3.2 Creation and training of echo state network

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).

3.2.1 Activation function

At each time step, the state vector $\mathbf{s}[n]$ (describing the activation level of every neuron) is updated according to

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

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 $(0<m\leq 1)$ is a parameter controlling the leak rate of each neuron. The output associated with the current state is given by

$$o[n] = \mathbf{W}_{\text{out}}\cdot\begin{bmatrix}\mathbf{s}[n]\\ 1\end{bmatrix}. \tag{14}$$

Note that the leak rate allows one to tune the short-term memory of each neuron and thus to change the dynamics of the reservoir. It has been observed [32, 33] that a crucial point to obtain good performance is to make the dominant time scale of the reservoir match

the dominant time scale of the input data. However, it is hard to say a priori what is the dominant time scale of the input data, so the usual approach is to find the optimal m by ranging through 0 and 1.

3.2.2 Network creation

The matrices $\mathbf{W}_{\text{in}}$ 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 [34]. An efficient trade-off has been found by setting the elements of $\mathbf{W}_{\text{in}}$ to -0.1 or $+0.1$ with equal probability. The reservoir connections must guarantee the echo state property [29]. 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 smaller than 1 (here we use a value of 0.9).

3.2.3 Network 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 output matrix should satisfy

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

where n_t is the number of time samples and $\hat{o}[n]$ is the desired output at time step n. Originally, this equation is solved in the least square sense. However, in order to achieve a better generalization to new situations, we applied here ridge regression and solved instead

$$\mathbf{W}_{\text{out}} = \arg\min_{\mathbf{W}} \left(\left\| \mathbf{W} \cdot \begin{bmatrix} \mathbf{s}[1] & \mathbf{s}[2] & \cdots & \mathbf{s}[n_t] \\ 1 & 1 & \cdots & 1 \end{bmatrix} - [\hat{o}[1]\ \ \hat{o}[2]\ \ \cdots\ \ \hat{o}[n_t]] \right\|^2 + \lambda \cdot \|\mathbf{W}\|^2 \right), \quad (16)$$

where λ is a regression parameter. The optimal value of λ is found via grid-search, by leaving out a validation set during training.

3.3 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

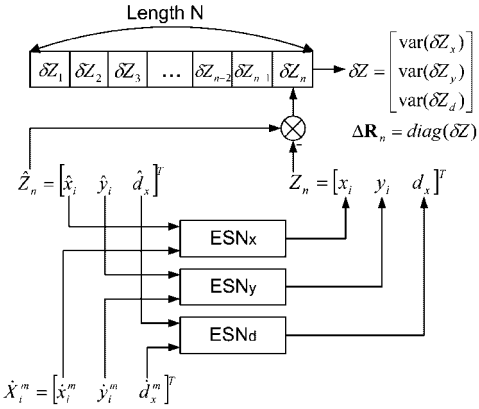

Fig. 5. Inputs and outputs of the ESNs (detail of the ESN box from Fig. 3).

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

$$\Delta \mathbf{R}_n = \text{diag}(\text{var}(\delta Z_x), \text{var}(\delta Z_y), \text{var}(\delta Z_d)), \quad (17)$$

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

IV. REAL-TIME FACE TRACKING UNDER ILLUMINATION VARIATION

This section presents the face tracking algorithm used in the proposed face tracking interaction control system. The proposed face tracking algorithm is a video color object tracker. Color is an efficient cue for face tracking, but skin color can easily depend on illumination and this can make face tracking fail [11]. To overcome this problem, the proposed face tracking algorithm utilizes YCrCb 3D color distribution model to effectively segment the skin color from other objects under illumination variation.

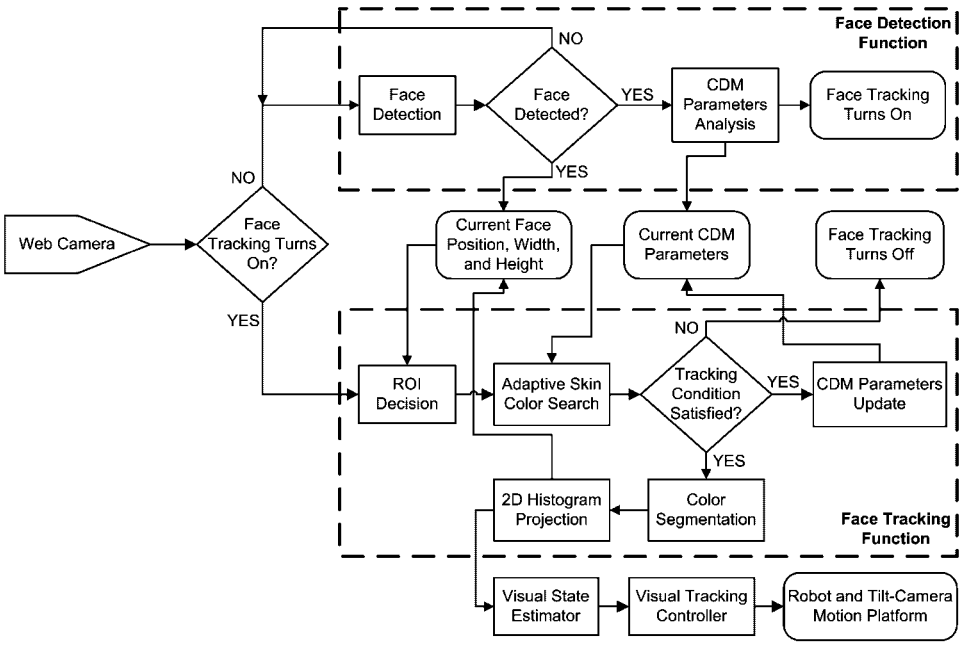

Fig. 6. The flowchart of the face tracking algorithm.

4.1 Face tracking algorithm overview

Fig. 6 shows the flowchart of the proposed face tracking algorithm, which consists of a face detection function and a face tracking function. In the initial state, the face detection function detects a face within the image frame and passes the position, width and height of the face to the face tracking function. With this information, the face tracking function determines a region of interest (ROI) to locate the face in the center of ROI. In order to obtain the optimal color distribution model (CDM) of the skin color in ROI for color segmentation, a novel adaptive skin color search (ASCS) method is proposed to accomplish this task. If the current skin color area extracted by ASCS method is within a certain proportion of ROI (termed as Tracking Condition, TC), the face tracking task will be regarded as a successful operation. Otherwise, the face tracking algorithm will be re-initialized into face detection stage.

If TC is satisfied, the optimal CDM parameters will be used to generate a binary image in ROI by color segmentation and will also be stored for use in the next tracking iteration. After obtaining a binary image of the skin color segmentation, a 2D histogram projection is generated to find the position, width and height of the face in ROI. Next, the information about the observed face in the current image will be stored for setting up the ROI in the next tracking iteration and will also be sent to the visual state estimator presented in Section III. Finally, the visual tracking controller presented in Section 2.2 controls the mobile robot and tilt-camera to track the user by using the estimation results from the visual state estimator.

4.2 Face detection function

If the face tracking function turns off, the face detection function will start to find the user's face. In our design, any existing face detection algorithm, *e.g.* [35, 36], can be applied to detect the human face in the current image. In this paper, the face detection algorithm proposed in [35] is adopted. When the user's face is detected, the information about the detected face in the current image will be stored for ROI decision in the face tracking function. Next, the CDM parameters (the color distribution thresholds) of user's skin color in the current image will be calculated by the Y, Cb, Cr histograms of the detected face and stored for use in the proposed ASCS method. Finally, the face tracking function enables to track the user's face in the next sampled image.

4.2.1 Proposed CDM parameters analysis method

Selecting a proper CDM is favorable not only for color segmentation, but also reduces the undesired error in the segmented result. To do so, we propose a statistical method to decide the proper threshold values for each color channel of the image. Let Y1(Y2) is the lower(upper) threshold of Y channel, Cr1(Cr2) is the

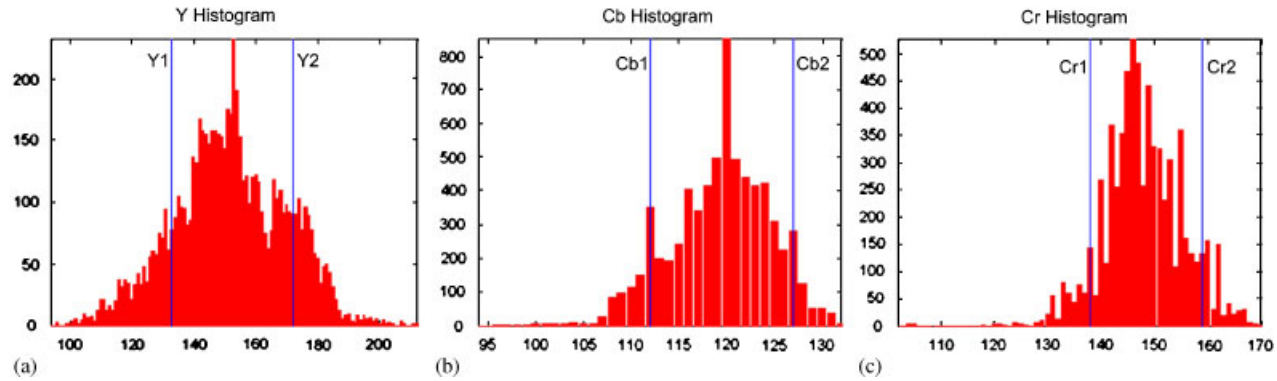

Fig. 7. The lower and upper thresholds of each color channel: (a) Y1 and Y2 of Y channel; (b) Cb1 and Cb2 of Cb channel; and (c) Cr1 and Cr2 of Cr channel.

lower(upper) threshold of Cr channel and Cb1(Cb2) is the lower(upper) threshold of Cb channel (see Figs 7(a), (b) and (c)). These threshold values are obtained by the following equations:

$$C_1 = \arg\min_i \left[\left(\sum_{i=C_{\min}}^{C_{\max}} C_{hist}(i) \right) - N_{total} \times S \right] > 0, \quad (18)$$

$$C_2 = \arg\min_i \left[\left(\sum_{i=C_{\max}}^{C_{\min}} C_{hist}(i) \right) - N_{total} \times S \right] > 0, \quad (19)$$

where C denotes one of three color channels, Y, Cb or Cr. C_{hist} is the histogram of C channel. S is a scale factor between 0 and 1. In this paper, S is set as 0.9. N_{total} is the total pixel number of a sub-search window (SSW, which will be defined in Section 4.4) in ROI. Fig. 7 shows an example of computing the threshold values for a local CDM of the skin color in a SSW located in ROI. When the threshold values for each color channel are founded, the color segmentation can be done in ROI by these threshold values such that

$$I_{skin}^{YCbCr}(x,y)|_{(x,y)\in ROI} = \begin{cases} 1, \text{ if the pixel (x,y)}\in\text{ROI satisfies } C_1 \le C \le C_2 \\ \qquad \text{for each color channel C} \\ 0, \text{ otherwise,} \end{cases} \quad (20)$$

where I_{skin}^{YCbCr} is a binary image of the skin color region segmented by CDM parameters (Y1,Y2,Cb1,Cb2,Cr1, Cr2). Note that the benefit of the proposed statistical scheme (18) and (19) is that the color distribution thresholds are adapted dependent on the current skin color information. Therefore, there is no limitation on the skin color by adopting the proposed statistical method.

4.3 Face tracking function

Suppose that the position, width and height of the user's face, $(x_i[0], y_i[0], d_x[0], d_y[0])$, in the image plane have been detected by the face detection function, and the face tracking function is enabled to track the detected face. The execution steps of the proposed face tracking function are described as follows:

1. ROI decision: The purpose of ROI is to shrink the face searching area and locate the face in the center area of ROI. The position, width and height of ROI, (ROI_$x_i[n]$, ROI_$y_i[n]$, ROI_$d_x[n]$, ROI_$d_y[n]$), are given by

$$\begin{aligned} \text{ROI\_}x_i[n] &= x_i[n-1], \\ \text{ROI\_}y_i[n] &= y_i[n-1], \\ \text{ROI\_}d_x[n] &= d_x[n-1]\times 2, \\ \text{ROI\_}d_y[n] &= d_y[n-1]\times 2, \end{aligned} \quad (21)$$

   where the initial position, width and height of ROI are defined such that

$$\begin{aligned} \text{ROI\_}x_i[0] &= x_i[0], \\ \text{ROI\_}y_i[0] &= y_i[0], \\ \text{ROI\_}d_x[0] &= d_x[0]\times 2, \\ \text{ROI\_}d_y[0] &= d_y[0]\times 2. \end{aligned} \quad (22)$$

   Please see Section 4.3.1 for more discussions on the advantages of ROI.
2. Adaptive skin color search (ASCS): Re-calculate the Y, Cb, Cr histograms of the detected face to find the optimal CDM parameters of the current skin color in ROI. The details of the proposed ASCS method are presented in Section 4.4.

(a) (b)

Fig. 8. The setting of the ROI: (a) the test image and (b) the image processed by color segmentation. The window of dotted line is ROI. The solid line in the ROI indicates the detected face.

3. Check the tracking condition (TC): Suppose that SA^*_{skin} denotes the skin color area in ROI segmented by the optimal CDM parameters obtained from ASCS, and TA_{ROI} denotes the total area of ROI. If the following condition

$$\text{(TC)} \quad TA_{ROI}/8 < SA^*_{skin} < TA_{ROI}/2$$

is satisfied, the face tracking event is successful, then go to both Step 4 and Step 5. Otherwise, the face tracking event is failed, and the face tracking algorithm is re-initialized into face detection stage to detect the user's face in the next sampled image.
4. CDM parameters update: Store the optimal CDM parameters obtained from ASCS for use in the next tracking iteration.
5. Color segmentation: Segment the skin color region in ROI by using the optimal CDM parameters obtained from ASCS. Please see Section 4.2 for the descriptions of the color segmentation method used in this paper.
6. 2D histogram projection: Find the position, width and height of the skin color region in ROI by using 2D histogram projection method [35]. Then the new information about the detected face, $(x_i[n], y_i[n], d_x[n], d_y[n])$, in the current sampled image will be updated for the next ROI decision and passed to the visual tracking control system for the task of face tracking interaction control. For more technical details of 2D histogram method, the interested readers may refer to [35].

4.3.1 Advantages of ROI

In order to obtain CDM parameters of the skin color in real-time, the idea of ROI is applied to achieve this goal. The advantages of using ROI are listed below:

1. Reduce the computation cost: Because the face tracking focuses inside the ROI, the computation outside the ROI is unnecessary.
2. Shrink the face tracking area: Fig. 8 shows that the areas containing the same color distribution of the current skin color will increase the search areas of face tracking. By searching only in ROI, the undesired areas outside ROI can be neglected for tracking the face efficiently and precisely.
3. Select proper thresholds for the skin color segmentation: In Section 4.2, two thresholds are assigned to each color channel for the skin color segmentation. If the thresholds are selected improperly, it may make the face tracking system not work. To resolve this problem, the proper threshold values are determined based on the information inside the ROI only.

Assume that the face is located at the center area of ROI during face tracking procedure (see Fig. 8(a)), then the proposed ASCS, which selects at least one SSW near the center area of the detected face, is exploited to get an optimal CDM of the current skin color for color segmentation. Furthermore, the position, width and height of current ROI are set according to the previous ones of the detected face area (see the expression (21)). By this way, the trail of face tracking will follow the previous successful face tracking result.

4.4 Proposed ASCS method

The purpose of the proposed ASCS method is to get the optimal CDM parameters of the current skin

color in ROI even as the color distribution of the skin color is changed due to the illumination variation. The execution steps of the proposed ASCS method are described as follows:

1. Sub-search window (SSW) decision: The purpose of SSW is to re-calculate CDM parameters of the current skin color in ROI. There are two principles to select SSW in ROI. First, the size of SSW is smaller than the detected face, but cannot be too small in order to prevent the loss of most color distribution information. Second, the position, width and height of each SSW are obtained according to the information about the previous detected face. For example, let k denote the number of SSW used in ROI, where $k \geq 1$. By the two principles of selecting SSW mentioned before, the width and height of each SSW, $(\mathrm{SSW}j\_d_x[n], \mathrm{SSW}j\_d_y[n])$ for $j=1\sim k$, are dependent on the information about previous detected face such that

$$\begin{aligned} &\mathrm{SSW}j\_d_x[n] = d_x[n-1]/2, \\ &\mathrm{SSW}j\_d_y[n] = d_y[n-1]/2, \quad \text{for } j=1\sim k. \end{aligned} \tag{23}$$

The position of the first SSW, $(\mathrm{SSW1}\_x_i[n], \mathrm{SSW1}\_y_i[n])$, is defined the same as the position of the previous detected face:

$$\begin{aligned} &\mathrm{SSW1}\_x_i[n] = x_i[n-1], \\ &\mathrm{SSW1}\_y_i[n] = y_i[n-1]. \end{aligned} \tag{24}$$

The position of other SSWs can be assigned to any different position around the center of SSW1 such that

$$\begin{aligned} &\mathrm{SSW}j\_x_i[n] = \mathrm{SSW1}\_x_i[n] + \Delta d_x^j, \\ &\mathrm{SSW}j\_y_i[n] = \mathrm{SSW1}\_y_i[n] + \Delta d_y^j, \end{aligned} \tag{25}$$

where $0<\Delta d_x^j<d_x[n-1]/4$, $0<\Delta d_y^j<d_y[n-1]/4$, for $j=2\sim k$, and $(\Delta d_x^p, \Delta d_y^p) \neq (\Delta d_x^q, \Delta d_y^q)$, for $p \neq q$, $p, q \in [1, k]$.

2. Re-calculate CDM parameters for each SSW: Each SSW is corresponding to a CDM of the skin color, which can be obtained by the proposed CDM analysis method presented in Section 4.2.1. Therefore, let $YCbCr|_{SSWj}$ denote the CDM parameters corresponding to SSW j, and $YCbCr|_{opt}^{-}$ is the previous optimal CDM parameters obtained in last sampled image. We then have $k+1$ groups of CDM parameters needed to be examined.
3. Search for the optimal CDM parameters of the current skin color in ROI: By the color segmentation method (20), each group of CDM parameters is able to obtain a corresponding skin color area in ROI such that

$$\begin{aligned} &SA_{skin}(YCbCr) \\ &\quad = \sum_x \sum_y I_{skin}^{YCbCr}(x, y)|_{(x,y)\in ROI}, \end{aligned} \tag{26}$$

where I_{skin}^{YCbCr} is a binary image of the skin color in ROI segmented using (20) by CDM parameters $YCbCr$. Thus, the optimal CDM parameters can be obtained by choosing the parameters maximizing the skin color area in ROI such that

$$YCbCr|_{opt}^{*} = \arg \max_{\substack{YCbCr|_{opt}^{-}, \\ YCbCr|_{SSWj}, \\ j=1\sim k}} SA_{skin}(\mathrm{YCbCr}). \tag{27}$$

Let $SA_{skin}^{*}(YCbCr|_{opt}^{*})$ denote the maximum skin color area in ROI obtained by the optimal CDM parameters $YCbCr|_{opt}^{*}$. The face tracking event then can be validated as succeeded or failed by tracking condition (TC) presented in Section 4.3.

Fig. 9 shows an example of the proposed ASCS method. In this example, four SSWs are used to gather four statistics CDMs on the center (SSW1), right (SSW2), bottom (SSW3) and left (SSW4) positions of the user's face, respectively. For each SSW, the proposed CDM analysis method, (18) and (19), is applied to obtain the corresponding CDM parameters. Using the previous optimal CDM parameters and four re-calculated ones, the new optimal CDM parameters then can be found by (27) and will be used in the next tracking iteration.

Remark 2. The originality of the proposed face tracking algorithm is twofold. First, the proposed CDM parameters analysis method presented in Section 4.2.1 can efficiently integrate a face detection function into a color-object-based face tracking algorithm without the restriction on the skin color due to the advantage of the proposed statistical scheme (18) and (19). Second, existing color-object based face tracking algorithms, such as CamShift algorithm [37], usually fail in the problem of rapid illumination variation. In contrast, the proposed real-time face tracking algorithm can efficiently overcome this problem due to the advantage of the proposed ASCS method presented in Section 4.4.

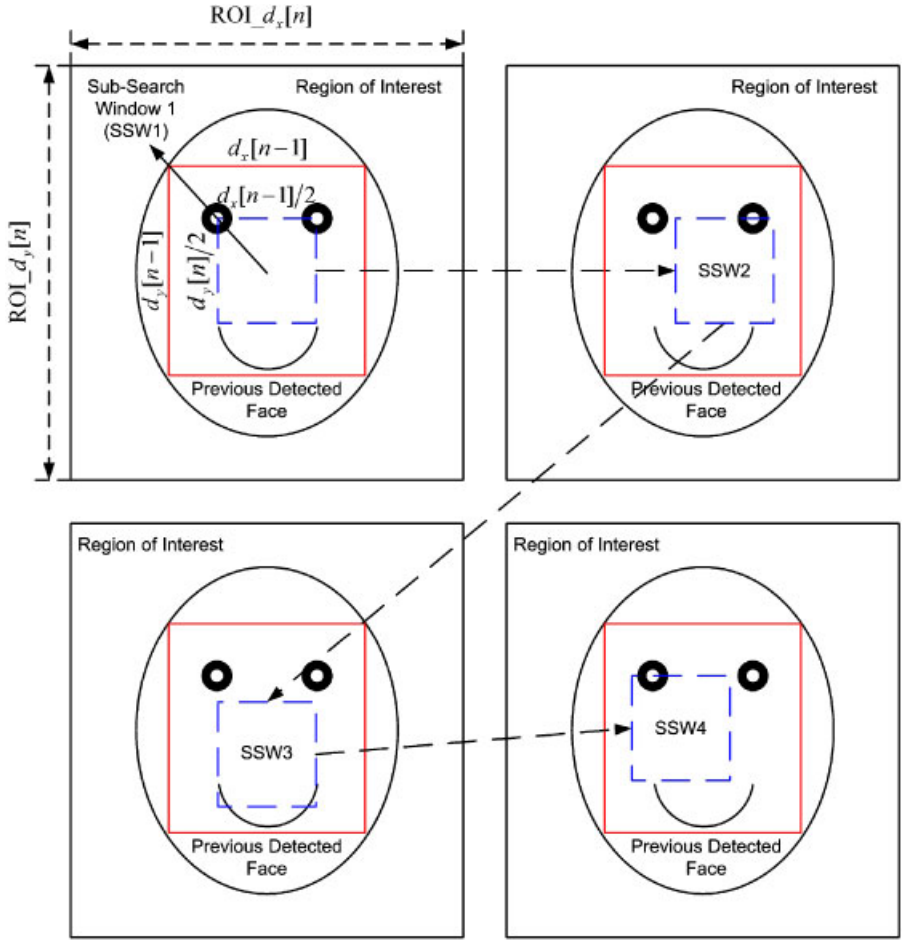

Fig. 9. An example of the proposed ASCS method. This example uses 4 SSWs to re-calculate the optimal CDM parameters of the current skin color in ROI.

V. SIMULATION AND EXPERIMENTAL RESULTS

5.1 Simulation setup

In order to evaluate the performance of the proposed visual state estimator, a simulation environment was setup using MATLAB. Fig. 10 shows the architecture of the simulation setup. In Fig. 10, 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)

$$RN = \begin{cases} K_n\sigma_1(0.5-\sigma_2), & if\ (\sigma_3<\rho) \\ (1+\sigma_1)(0.5-\sigma_2), & otherwise \end{cases} \tag{28}$$

where $K_n>1$ is the noise gain; $\sigma_i \in [0, 1]$, $i=1\sim3$, are three random signals with uniform distribution; and $\rho \in [0, 1]$ is a constant threshold value. Expression (28) indicates that the intensity of the noise is time-varying and dependent on a random condition. If the condition $(\sigma_3<\rho)$ is satisfied, then the random noise will have large noise gain; otherwise the random noise 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 random noise usually happens during practical visual tracking process of the mobile robot, since the intensity of the observation uncertainty usually is position-dependent and light-dependent.

In the following, a visual state estimator is utilized to filter the random noise 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

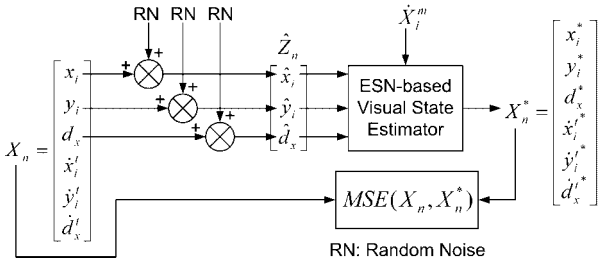

Fig. 10. Simulation setup for the performance evaluation of the visual state estimator.

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.

5.2 Simulation results

There are three visual state estimators used to compare the performance: Kalman filter (KF), self-tuning Kalman filter using linear regression (STKF-LR) [25], 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 increases from 0 to 1, the average MSE measurements also increase apparently. 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 compared with the other two estimators. Note that STKF-LR uses the measurement offset for the computation of observation variance. Please refer to [25] 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 that 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.

5.3 Experiment setup

Fig. 11 shows the experimental mobile robot, *RoLA*, used in the experiments. *RoLA* stands for robot of living aid, designed to provide immediate medical care for the elderly. 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 the 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 (the pan angle of the camera is constant). Fig. 12 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 80 ms including face tracking algorithm, estimator and controller computations. Thus, the overall tracking system can track the user's face in real-time.

5.3.1 Experimental results of face tracking under illumination variation

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 in the simulations.

Fig. 13 shows the experimental results of the implemented visual tracking control system given in Fig. 12. Figs 13(a1)–(a3) illustrate recorded pictures from a digital video (DV) camera, and Figs 13(b1)–(b3) show the corresponding pictures recorded by the on-board USB camera. In Figs 13(a1)–(a3), 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 13(b1–b3), the person's face suddenly became darker due to a rapid decrease in illumination. In such situations, the proposed face tracking algorithm works to adapt to the change in the skin color. However, the update of new skin color model results rapid random noises in the observation of system state. To overcome this problem, the ESNs work to provide a stable output even when the observation contains rapid random noises. Therefore, the robot still estimated and tracked the person's face in the image plane stably. Note that the image sequence shown in Fig. 13 is about 1 second.

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

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**

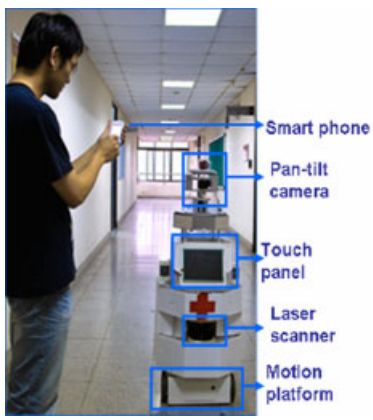

Fig. 11. An elder-care mobile robot, *Rola*, used in the experiments.

Subsequently, the tracked person was walking in the environment with a low illumination. Fig. 14 illustrates the experimental results under this situation. As shown in Figs 14(a1)–(a3) and Figs 14(b1)–(b3), the environment and person's face suddenly became lighter due to a rapid increase in illumination, and the proposed face tracking algorithm still works to obtain an optimal skin color model to keep tracking the detected face. In this case, the proposed face tracking algorithm and ESNs both have stable outputs even when the illumination is suddenly changed. Therefore, the experimental results show that the proposed face tracking algorithm and ESN-based visual state estimator can track the face under illumination variation and remove the undesired noise in observation, respectively. Note that the image sequence shown in Fig. 14 is about 2 seconds.

Fig. 15 presents the recorded experimental results of face tracking under illumination variation. Figs 15(a)–(c) show a comparison between the observed error states (the dotted lines with spikes) and the corresponding ESN outputs (the solid lines), and the estimated ones (denoted by x_e^*, y_e^*, and d_e^*). From Figs 15(a)–(c), we see that the observation noise caused

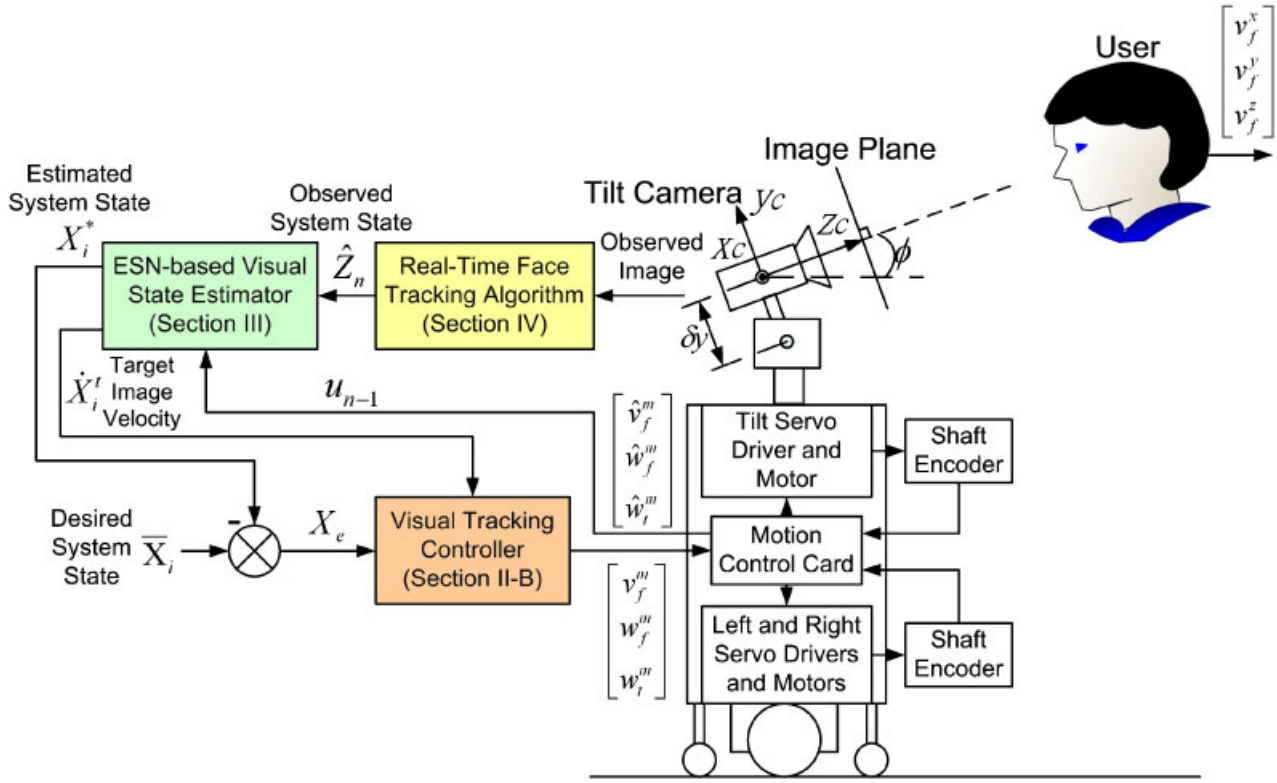

Fig. 12. Block diagram of the implemented visual tracking control system, which includes the proposed ESN-based visual state estimator.

Fig. 13. Experimental results. (a1)–(a3): Image sequence recorded from a DV camera. (b1)–(b3): Corresponding image sequence recorded from on-board USB camera. In the pictures (b1)–(b3), the green window indicates the observation, and the blue window is the corresponding output of ESNs.

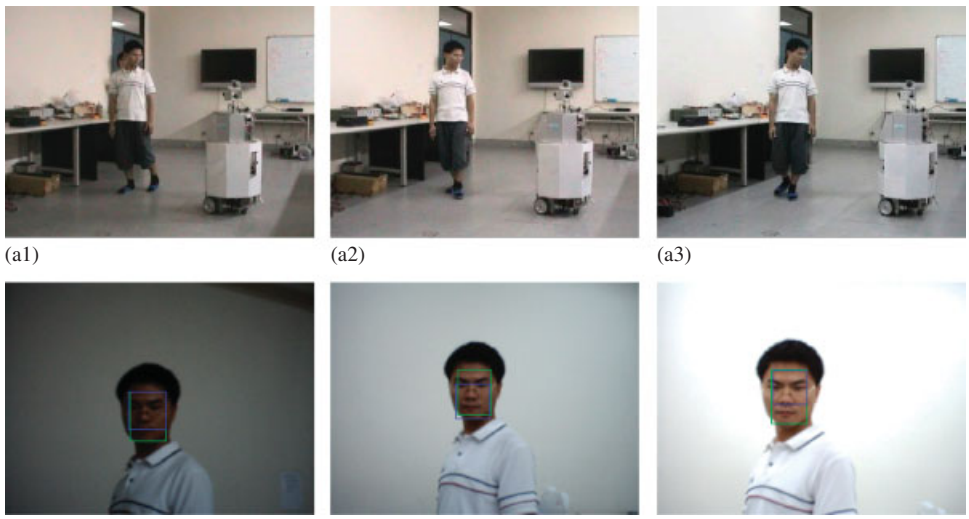

Fig. 14. Experimental results. (a1)–(a3): Image sequence recorded from a DV camera. (b1)–(b3): Corresponding image sequence recorded from on-board USB camera. In the pictures (b1)–(b3), the green window indicates the observation, and the blue window is the corresponding output of ESNs.

by the effect of illumination variation is removed efficiently by utilizing the proposed ESN-based visual state estimator. Fig. 15(d) shows the control velocities of both mobile robot and tilt camera.

Remark 3. In Fig. 15(a), we observe that the ESN output of tracking error x_e is bounded in $(-60, 60)$. This is caused by that the reference (or teacher output) of training data used in training process is bounded in the same range. The main reason to do this is as follows. Empirically, a large tracking error value leads a large control velocity output. Because there is usually a velocity limitation on the motion of a practical mobile robot ($\leq 20\,\text{cm/s}$ in the experiments), a mechanism is required to guarantee that the control velocity output satisfies the velocity limitation. This can be achieved by bounding the output of system estimator, which usually leads a bounded command output of system controller. With ESN, this can be achieved in a strict way by post-processing the output given by (14) with a bounding function or by putting a hard limit on the weights of the readout matrix $\mathbf{W}_{\text{out}}$ (as the activity of each neuron is bounded due to the hyperbolic tangent non-linearity). However, it is usually sufficient to only use a bounded training dataset when training the ESN, and this is the approach we use here. The boundary of the ESN output can then be adjusted by the boundary of the training dataset.

5.3.2 Experimental results of face tracking with occlusion robustness

Fig. 16 shows the recorded images of the mobile robot interacting with a walking person in the experiment of face tracking with temporary occlusion. Figs 16(a1)–(a3) and 16(b1)–(b3), respectively, show the recorded photos of the experimental scenario by a DV camera and the on-board USB camera. In the experiment, the person walked around in the room, and the mobile robot kept tracking the person's face by the tilt camera (Figs 16(a1) and 16(b1)). When the person was walking, another person passed between the tracked person and the robot temporarily (Fig. 16(a2)). Thus, in Fig. 16(b2), the person's face was temporarily fully blocked by the passing person. Based on the proposed self-tuning Kalman filter algorithm, the propagation information will dominate the estimation results in this situation even if the target is fully unobservable. Therefore, the visual state estimator still estimated the positions and velocities of the person's face in the image plane successfully even during full occlusion conditions (Figs 16(a3) and (b3)). Therefore, based on the above experiments of face tracking with illumination variation and occlusion, the robust estimation performance of the proposed face tracking interaction control system is verified. Several video clips of mobile robot visual tracking experimental results are available online in [38].

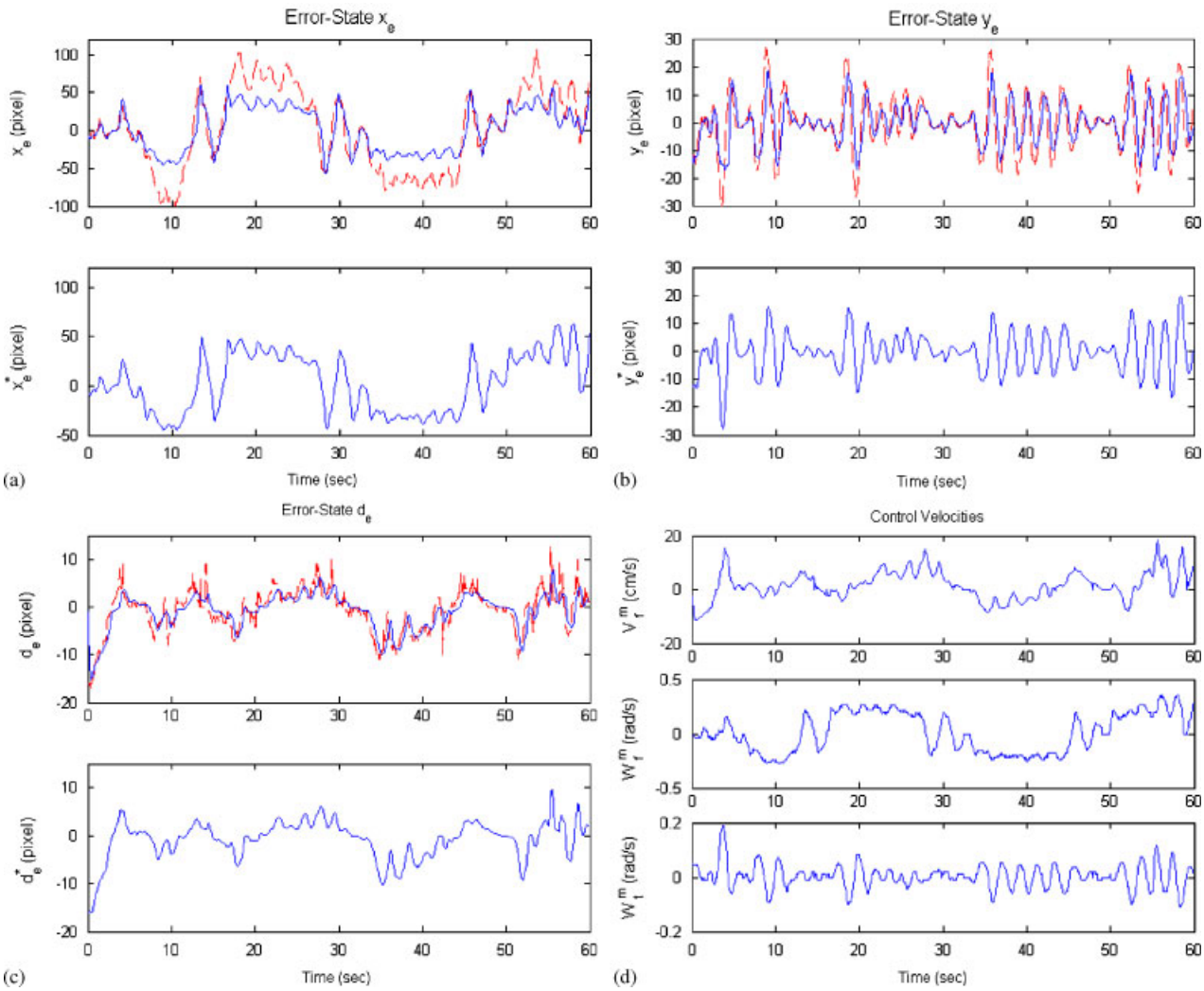

Fig. 15. Experimental results of the proposed visual tracking controller combined with ESN-based visual state estimator. The observed tracking errors (a) x_e, (b) y_e, (c) d_e (dotted lines) compared with the corresponding ESN outputs (solid lines), and the estimated tracking errors (a) x_e, (b) y_e, (c) d_e. (d) Command velocities of mobile robot and tilt camera.

5.4 Robustness property of ESN noise filter

Although conventional Kalman filters have a little robustness against some special disturbances, the optimality of the Kalman filter cannot be guaranteed under the condition that the statistics of the observation noise is non-Gaussian and colored. In order to overcome this problem, the proposed self-tuning Kalman filter employs an ESN noise filter to filter the colored observation noise and output the filtered observation with Gaussian white noise, which satisfies the optimal condition of the Kalman filter. In other words, the robustness of the conventional Kalman filter can be enhanced without any modification to handle colored observation noise by combining an ESN noise filter with a self-tuning algorithm. This section utilizes the colored random noise defined in (28) and three general types of noises, the Gaussian white noise, uniform white noise, and colored noise, to validate the noise filtering capability of ESN noise filter. The colored noise $\xi(k)$ is modeled by a shaping filter with a Gaussian white noise such that

$$\xi(k+1)=\psi\xi(k)+\varepsilon(k), \tag{29}$$

where $\psi=0.95$ is the transition matrix of the shaping filter, and $\varepsilon(k)\in[-1,1]$ is a Gaussian white noise with zero mean and variance 1.

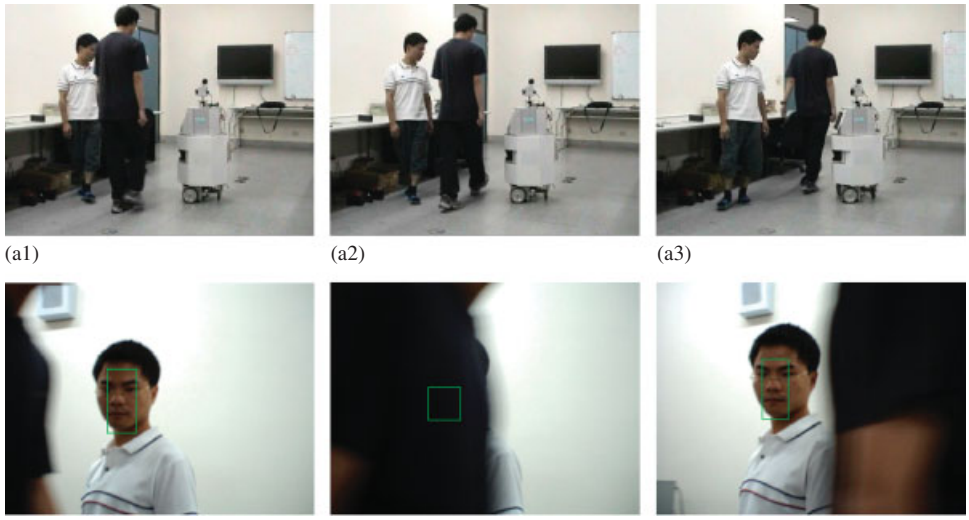

Fig. 16. Experimental results. (a1)–(a3): Image sequence recorded from a DV camera. (b1)–(b3): Corresponding image sequence recorded from on-board USB camera. In the pictures (b1)–(b3), the green window indicates the estimation result of the proposed ESN-based visual state estimator.

One of the three independent ESNs used in the experiments is used to demonstrate the noise filtering capability. The simulation results are shown in Fig. 17, which presents the error distribution of ESN input and output data. In Fig. 17, the red curves are the fitting results of the actual distribution by a polynomial curve fitting method. Figs 17(a), (b), (c), and (d) show the results of Gaussian white noise, uniform white noise, shaping filter colored noise (29), and colored random noise given by (28), respectively. In Fig. 17, x denotes the normalized random variable of error distribution, and $p(x)$ is the corresponding probability density function. From Fig. 17, we have the following findings.

1. The error distributions of all ESN output data (the filtered observation) are close to Gaussian white noise.
2. In Fig. 17(a), the error distributions of ESN input and output data are all Gaussian. This means that the optimal condition of the Kalman filter will not be changed by directly using the output measurement of the ESN noise filter.
3. Since the colored random noise (28) was used in the training data set, the ESN noise filter provides the best filtering results (the minimum variance of output error distribution) compared to the others. This result also explains why the proposed STKF-ESN has stronger robustness than the general KF and STKF-LR shown in Table II.
4. Because the colored random noise (28) is generated by uniform white noise, the trained ESN noise filter also has good performance to filter uniform white noise as shown in Fig. 17(b).

Therefore, based on the above findings, the conventional Kalman filter can guarantee to provide the optimal estimation for ESN output measurement, which implies that the robustness of the conventional Kalman filter can be improved to handle Gaussian white noise and colored random noise as we expected.

VI. CONCLUSION

In this paper, a robust visual tracking control system for a wheeled mobile robot is proposed based on ESN-based self-tuning Kalman filter algorithm and visual tracking control techniques. This design can be applied to several visual tracking applications, such as visual tracking control, visual surveillance, and visual navigation, etc. for a wheeled mobile robot to interact with a target in the image plane. One of the main contributions of the proposed ESN-based self-tuning Kalman filter is that the robustness of the original Kalman filter can be improved against not only white noise, but also colored noise without any modification. In order for the robot to interact with the user by visual tracking control, the proposed system is combined with a real-time

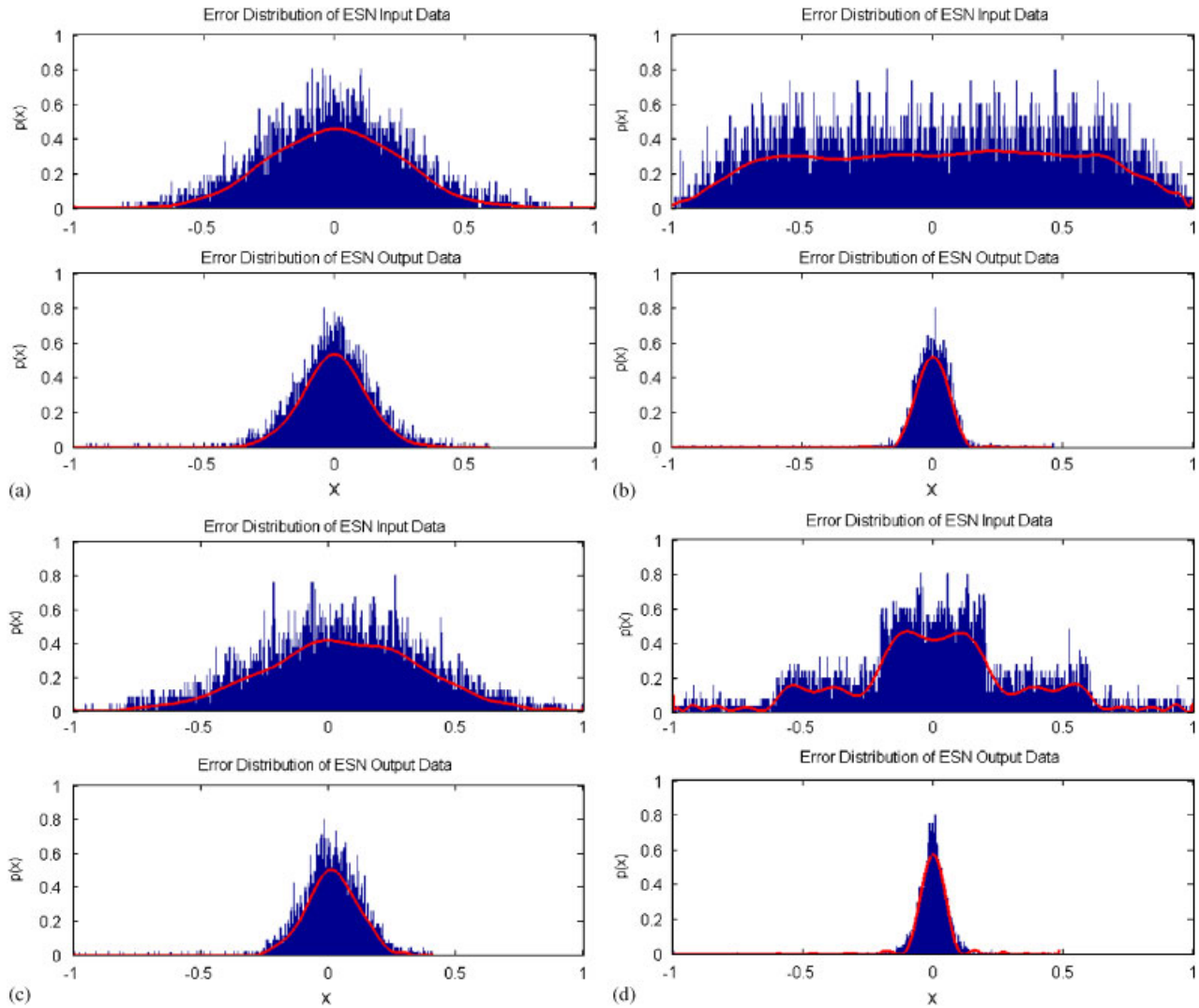

Fig. 17. Error distribution of ESN input and ESN output data. ESN Input data with (a) Gaussian white noise, (b) uniform white noise, (c) colored noise modeled by the shaping filter (29), and (d) colored noise given by (28). It is clear that the error distributions of all ESN output data are close to Gaussian white noise. Note that all distributions were normalized.

face tracking algorithm to detect and track the user's face in the image plane. One merit of the proposed real-time face tracking algorithm is that the proposed ASCS method can overcome the problem of illumination variation during visual tracking processing. By combining the proposed real-time face tracking algorithm with the robust visual tracking control system, the performance of the mobile robot face tracking interaction control is thus enhanced to cope with the observation uncertainty caused by colored noise, illumination variation and occlusion. This advantage is very useful in robotic applications, since the observation uncertainty usually varies with the conditions of target motion and working environment. Computer simulations show that the proposed ESN-based visual state estimator provides high robustness against the colored observation noise with time-varying intensity by comparing with the conventional Kalman filter and the linear regression based self-tuning Kalman filter. Moreover, experimental results also verify the tracking performance of the proposed mobile robot face tracking interaction control system in a practical environment under light-varying and occlusion conditions.

REFERENCES

1. Mariottini, G. L., G. Oriolo, and D. Prattichizzo, "Image-based visual servoing for nonholonomic

mobile robots using epipolar geometry," *IEEE Trans. Robot.*, Vol. 23, No. 1, pp. 87–100 (2007).
2. Chen, J., W. E. Dixon, D. M. Dawson, and M. McIntyre, "Homography-based visual servo tracking control of a wheeled mobile robot," *IEEE Trans. Robot.*, Vol. 22, No. 2, pp. 407–416 (2006).
3. Coulaud, J-B., G. Campion, G. Bastin, and M. D. Wan, "Stability analysis of a vision-based control design for an autonomous mobile robot," *IEEE Trans. Robot.*, Vol. 22, No. 5, pp. 1062–1069 (2006).
4. Malis, E. and S. Benhimane, "A unified approach to visual tracking and servoing," *J. Robot. Auton. Syst.*, Vol. 52, No. 1, pp. 39–52 (2005).
5. Freda, L. and G. Oriolo, "Vision-based interception of a moving target with a nonholonomic mobile robot," *J. Robot. Auton. Syst.*, Vol. 55, No. 6, pp. 419–432 (2007).
6. Han, Y. and H. Hahn, "Visual tracking of a moving target using active contour based SSD algorithm," *J. Robot. Auton. Syst.*, Vol. 53, No. 3–4, pp. 265–281 (2005).
7. Chou, Y.-T. and P. Bajcsy, "Toward face detection, pose estimation and human recognition from hyperspectral imagery," *Automated Learning Group*, *National Center for Supercomputing Applications*, *Technical Report* (2004).
8. Sobottka, K. and I. Pitas, "Face localization and feature extraction based on shape and color information," *Proc. IEEE Int. Conf. Image Process.*, Lausanne, Switzerland, pp. 483–486 (1996).
9. Jang, G.-J. and I.-S. Kweon, "Robust object tracking using an adaptive color model," *Proc. IEEE Int. Conf. Robot. Autom.*, Seoul, Korea, pp. 1677–1682 (2001).
10. Stern, H. and B. Efros, "Adaptive color space switching for face tracking in multi-colored lighting environments," *Proc. 5th IEEE Int. Conf. Auton. Face Gesture Recognit.*, Washington, DC, pp. 236–241 (2002).
11. Lee, Y.-B., B.-J. You, and S.-W. Lee, "A real-time color-based object tracking robust to irregular illumination variation," *Proc. IEEE Int. Conf. Robot. Autom.*, Seoul, Korea, pp. 1659–1664 (2001).
12. Yang, M.-H., D. J. Kriegman, and N. Ahuja, "Detecting faces in images: a survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 24, No. 1, pp. 34–58 (2002).
13. Chaumette, F. and S. Hutchinson, "Visual servo control part I: basic approaches [tutorial]," *IEEE Robot. Autom. Mag.*, Vol. 13, No. 4, pp. 82–90 (2006).
14. Chaumette, F. and S. Hutchinson, "Visual servo control part II: advanced approaches [tutorial]," *IEEE Robot. Autom. Mag.*, Vol. 14, No. 1, pp. 109–118 (2007).
15. Ma, Y., J. Košecká, and S. S. Sastry, "Vision guided navigation for a nonholonomic mobile robot," *IEEE Trans. Robot. Autom.*, Vol. 15, No. 3, pp. 521–536 (1999).
16. Zhang, H. and J. P. Ostrowski, "Visual motion planning for mobile robots," *IEEE Trans. Robot. Autom.*, Vol. 18, No. 2, pp. 199–208 (2002).
17. Fang, Y., W. E. Dixon, D. M. Dawson, and P. Chawda, "Homography-based visual servo regulation of mobile robots," *IEEE Trans. Syst. Man Cybern. Part B-Cybern.*, Vol. 35, No. 5, pp. 1041–1049 (2005).
18. Xu, L.-Q. and D. C. Hogg, "Neural networks in human motion tracking – An experimental study," *Image Vis. Comput.*, Vol. 15, No. 8, pp. 607–615 (1997).
19. Ghosh, B. K. and E. P. Loucks, "A perspective theory for motion and shape estimation in machine vision," *SIAM J. Control Optim.*, Vol. 33, No. 5, pp. 1530–1559 (1995).
20. Ghosh, B. K., H. Inaba, and S. Takahashi, "Identification of Riccati dynamics under perspective and orthographic observations," *IEEE Trans. Autom. Control*, Vol. 45, No. 7, pp. 1267–1278 (2000).
21. Chen, X. and H. Kano, "A new state observer for perspective systems," *IEEE Trans. Autom. Control*, Vol. 47, No. 4, pp. 658–663 (2002).
22. Dixon, W. E., Y. Fang, D. M. Dawson, and T. J. Flynn, "Range identification for perspective vision systems," *IEEE Trans. Autom. Control*, Vol. 48, No. 12, pp. 2232–2238 (2003).
23. Chitrakaran, V., D. M. Dawson, W. E. Dixon, and J. Chen, "Identification of a moving object's velocity with a fixed camera," *Automatica*, Vol. 41, No. 3, pp. 553–562 (2005).
24. Tsai, C.-Y. and K.-T. Song, "Dynamic visual tracking control of a mobile robot with image noise and occlusion robustness," *Image Vis. Comput.*, Vol. 27, No. 8, pp. 1007–1022 (2009).
25. Tsai, C.-Y., K.-T. Song, X. Dutoit, H. Van Brussel, and M. Nuttin, "Robust mobile robot visual tracking control system using self-tuning Kalman filter," *Proc. IEEE Int. Symp. Comp. Intell. Robot. Autom.*, Jacksonville, Florida, pp. 161–166 (2007).
26. Schutter, J. D., J. D. Geeter, T. Lefebvre, and H. Bruyninckx, "Kalman filters: a tutorial," *Journal A*, Vol. 40, No. 4, pp. 52–59 (1999).

27. Korniyenko, O. V., M. S. Sharawi, and D. N. Aloi, "Neural network based approach for tuning Kalman filter," *Proc. IEEE Int. Conf. Electron./Inf. Technol.*, Lincoln, Nebraska, pp. 1–5 (2005).
28. Xiong, S.-S. and Z.-Y. Zhou, "Neural filtering of colored noise based on Lalman Filter structure," *IEEE Trans. Instrument. Measure*, Vol. 52, No. 3, pp. 742–747 (2003).
29. Jaeger, H. "The 'Echo State' approach to analysing and training recurrent neural networks," *German National Research Center for Information Technology*, *Technical Report* (2001).
30. Gibson, J. D., B. Koo, and S. D. Gray, "Filtering of colored noise for speech enhancement and coding," *IEEE Trans. Signal Process.*, Vol. 39, No. 8, pp. 1732–1742 (1991).
31. Tsai, C.-Y. and K.-T. Song, "Visual tracking control of a wheeled mobile robot with system model and velocity quantization robustness," *IEEE Trans. Control Syst. Technol.*, Vol. 17, No. 3, pp. 520–527 (2009).
32. Jaeger, H., M. Lukoševičius, D. Popovici, and U. Siewert, "Optimization and applications of Echo state networks with leaky integrator neurons," *Neural Netw.*, Vol. 20, No. 3, pp. 335–352 (2007).
33. Schrauwen, B., J. Defour, D. Verstraeten, and J. Van Campenhout, "The introduction of time scales in reservoir computing, applied to isolated digits recognition," *Proc. Springer Int. Conf. Artif. Neural Netw.*, *Part I—Lec. Notes Comput. Sci.*, Porto, Portugal, pp. 471–479 (2007).
34. Jaeger, H., "A tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the 'echo state network' approach," *German National Research Center for Information Technology*, *Technical Report* (2002).
35. Garcia, C. and G. Tziritas, "Face detection using quantized skin color regions merging and wavelet packet analysis," *IEEE Trans. Multimedia*, Vol. 1, No. 3, pp. 264–277 (1999).
36. Song, K.-T. and W.-J. Chen, "Face recognition and tracking for human-robot interaction," *Proc. IEEE Int. Conf. Syst. Man Cybern.*, Hague, Netherlands, pp. 2877–2882 (2004).
37. Bradski, G. R. and S. Clara, "Computer vision face tracking for use in a perceptual user interface," *Intell. Technol. J.*, Vol. 2, No. 2, pp. 1–15 (1998).
38. The experiment video website, http://isci.cn.nctu.edu.tw/video/RVTS_ESN/.

Chi-Yi Tsai was born in Kaohsiung, Taiwan, Republic of China in 1978. He received the B.S. and M.S. degrees in Electrical Engineering from National Yunlin University of Science and Technology, Yunlin, Taiwan, in 2000 and 2002, respectively. He received the Ph.D. degree in Electrical and Control Engineering, at the National Chiao Tung University, Taiwan, in 2008. From September 2007 to August 2009, he was a software engineer in software R&D department of software R&D division, ASUSTek Computer Incorporation. From September 2009 to January 2010, he was an assistant researcher in Chung-Shan Institute of Science & Technology, Taiwan. He is currently an assistant professor in department of electrical engineering, Tamkang University, Taiwan. His research interests include image processing, color enhancement processing, visual tracking control for mobile robots, visual servoing and computer vision.

Xavier Dutoit was born in Switzerland in 1981. He received his M.S. degree in Communications Systems from the Ecole Polytechnique Federale de Lausanne, Switzerland, in 2004. From 2005 to 2010, he was conducting research in machine learning and mobile learning robots at the Katholieke Universiteit of Leuven, Belgium, where he received his Ph.D. in engineering in 2010.

Kai-Tai Song was born in Taipei, Taiwan, in 1957. He received the B.S. degree in Power Mechanical Engineering from National Tsing Hua University in 1979 and the Ph.D. degree from the Katholieke Universiteit Leuven, Belgium in 1989. He was with Chung Shan Institute of Science and Technology from 1981 to 1984. Since 1989 he has been on the faculty and is currently a Professor in the Department of Electrical Engineering, National Chiao Tung University, Taiwan. From 2007 to 2009, he was the Associate Dean of Research & Development Office of the university. Since 2009, he also serves as the Director of Institute of Electrical and Control Engineering, National Chiao Tung University.

He served as the Chair of IEEE Robotics & Automation Chapter, Taipei Section in the term of 1999. His areas of research interest include mobile robotics, image processing, visual tracking, embedded systems, and mechatronics.

Prof. Hendrik Van Brussel (1944), is full Professor of Mechatronics and Automation at the Faculty of Engineering, Katholieke Universiteit Leuven (K. U. Leuven), Belgium, and Chairman of the Division of Production Engineering, Machine Design and Automation (PMA), Department of Mechanical Engineering.

He was a pioneer in robotics research in Europe and an active promoter of the mechatronics idea as a new paradigm in machine design. He has published extensively on different aspects of robotics, mechatronics and flexible automation. His present research interest is also shifting towards holonic manufacturing systems, behaviour based robots, and micro and precision engineering, including microrobotics.

He is a Fellow of SME and IEEE. He was President of CIRP (International Academy for Production Engineering). He is a Member of the Royal Flemish Academy of Belgium for Sciences and Arts, and Foreign Member of the Royal Swedish Academy of Engineering Sciences (IVA). He is President of euspen (European Society for Precision Engineering and Nanotechnology).

Marnix Nuttin was born in 1969 in Belgium. He obtained his engineering degree from the Katholieke Universiteit Leuven in 1992, and his Ph.D. in 1998. He has been conducting research on mobile learning robots from 1992 to 2008 in the Department of Mechanical Engineering. He developed learning techniques for Programming by Demonstration and adaptive shared autonomy for wheelchair control. He participated in the ITEA AMBIENCE project (context-aware environments for ambient services). In FP6 he was principle investigator of the projects MOVEMENT (Modular Versatile Mobility Enhancement Technology), MAIA (Non Invasive Brain Interaction with Robots) and DYNAVIS (Dynamically Reconfigurable Quality Control). He is currently affiliated with the Faculty of Engineering.