ARTICLE IN PRESS

Neurocomputing ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Extreme learning machine towards dynamic model hypothesis in fish ethology research

Rui Nian [a], Bo He [a,*], Bing Zheng [a], Mark van Heeswijk [b], Qi Yu [b], Yoan Miche [b], Amaury Lendasse [b,c,d,e]

[a] College of Information Science and Engineering, Ocean University of China, 266003 Qingdao, China
[b] Department of Information and Computer Science, Aalto University, FI-00076 Aalto, Finland
[c] IKERBASQUE, Basque Foundation for Science, 48011 Bilbao, Spain
[d] Computational Intelligence Group, Computer Science Faculty, University of the Basque Country, Paseo Manuel Lardizabal 1, Donostia/San Sebastián, Spain
[e] Arcada University of Applied Sciences, 00550, Helsinki, Finland

ARTICLE INFO

Article history:
Received 29 August 2012
Received in revised form
21 February 2013
Accepted 20 March 2013

Keywords:
Extreme learning machine
Fish ethology
Video surveillance system
Dynamic state space
Color distribution
Object recognition

ABSTRACT

In this paper, we present one dynamic model hypothesis to perform fish trajectory tracking in the fish ethology research and develop the relevant mathematical criterion on the basis of the Extreme Learning Machine (ELM). It is shown that the proposed scheme can conduct the non-linear and non Gaussian tracking process by multiple historical cues and current predictions – the state vector motion, the color distribution and the appearance recognition, all of which can be extracted from the single-hidden layer feedforward neural network (SLFN) at diverse levels with ELM. The strategy of the hierarchical hybrid ELM ensemble then combines the individual SLFN of the tracking cues for the performance improvements. The simulation results have shown the excellent performance in both robustness and accuracy of the developed approach.

1. Introduction

The 21 century is an Era of the 'Ocean'. There has been a growing trend towards the deployment of the ocean blueprint all over the world. Fish ethology, an emerging discipline to explore the inherent nature of the movement, behaviors and activities for either wild or cultured fish, has shown great prospects in the aquaculture, fisheries, and other marine related surveys and applications [1–4].

Among a mass of fish behavior descriptions, the fish trajectory tracking is essential and fundamental. Tracking the fish can be extremely complex due to the random fish movements, all kinds of shape variations, the non rigid or articulated nature, the partial and full occlusions, the scene illumination changes, the multiple viewpoints, the poor image quality, the projection of the 3D world on 2D images, the real-time processing requirements, and so on.

So far, object tracking in literature mainly focus on classical approaches such as the background subtraction, the inter-frame difference, the optical flow computation, the Kalman filtering, the particular filtering, the mean-shift algorithms, etc., and the primary differences come from the type of the object representation, the feature extraction, the motion modeling, the shape and appearance, and the context that the tracking is performed [5–8].

In practice, the fish activities often correspond to a complicated, nonlinear, and non-Gaussian dynamic system, and the Bayesian estimation theory can be seen to be a philosophically optimal solution [9–11]. In cases that there is no prior knowledge available for the overall functional form of probability distribution beforehand, the scheme of the dynamic model approximation from the observations is the central concern for the fish ethology research.

In the context of the machine learning, Artificial Neural Networks (ANN) has been playing the dominant roles due to benefits on generalization, flexibility, nonlinearity, fault tolerance, self organization, adaptive learning, and computation in parallel, while the bottlenecks such as the overfitting, local minima, time consuming etc., can probably restrict the scalability in the conventional implementations [12–18]. Recently, the Extreme learning machine (ELM) has made a great breakthrough in the single-hidden layer feedforward neural network (SLFN) instead of the classical gradient-based algorithms [19,20]. The achievements of ELM tend to provide better generalization performance than the traditional approaches, and seek straightforward solutions mathematically with inspiring abilities such that the hidden node parameters can be randomly chosen and the output weights can be analytically determined at extremely fast learning speed and

* Corresponding author.
E-mail address: bhe@ouc.edu.cn (B. He).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.03.054

Please cite this article as: R. Nian, et al., Extreme learning machine towards dynamic model hypothesis in fish ethology research, Neurocomputing (2013), http://dx.doi.org/10.1016/j.neucom.2013.03.054

with least human intervene. By far, ELM has not only developed for the conventional SLFN [21,22], but also extended to the generalized SLFN that need not be the neuron alike [23,24]. There are a great many ELM variations that have been proposed and have led to the state-of-the-art results in many applications both for the regression problem and the pattern recognition problem [25–32]. Random hidden layer feature mapping based ELM improves the stability in the calculation of the output weights according to the ridge regression theory [25–27]. The Kernel based ELM makes use of the corresponding kernel instead of the hidden layer feature mapping itself, and the dimensionality of the hidden layer feature space needs not be specified either [25,28,29]. The fully complex ELM can use the fully complex activation function directly with the universal approximation capability [20]. The incremental ELM (I-ELM) shows an efficient and practical way to construct the incremental feed-forward network with a wide type of activation functions, where the hidden nodes can be added one by one [23,24]. The online sequential ELM (OS-ELM) can learn the training data sequentially not only one-by-one but also chunk by chunk and discard the observations as soon as the learning procedure has already been done [20]. The optimally-pruned ELM (OP-ELM) starts with a large network and then eliminates the hidden nodes that have low relevance to the learning [33,34]. ELM ensembles are widely used to improve single network's performance with a plurality consensus scheme [30–32].

In this paper, we develop a scheme of dynamic model hypothesis by means of ELM learning algorithm for fish ethology research. The rest of the paper is organized as follows: In Section 2, the background of the Bayesian estimation theory will be briefly introduced. In Section 3, the basics of the ELM are outlined. In Section 4, the dynamic model with ELM is developed in detail. In Section 5, the simulation and discussion will be stated in support of the developed scheme. Section 6 comes to the conclusions.

2. Bayesian estimation

In principle, for the fish trajectory tracking, the Bayesian sequential estimation can seek an optimal model [9–11]. The general dynamic model can be considered as the state transition and the state measurement,

$$X_t = f(X_{t-1}, U_{t-1}), \quad Y_t = h(X_t, R_t) \tag{1}$$

where t is the time index, X_t refers to the state variable of the fish propagated by the possibly nonlinear process model f over time, such as the position, velocity, etc., h is the observation model mapping the state variable X_t to the corresponding observation variable Y_t, U_t and R_t are respectively the process noise and the measurement noise that are roughly supposed as white Gaussian noise. The state prediction function is formulated as

$$p(X_t|Y_{1:t-1}) = \int p(X_t|X_{t-1})p(X_{t-1}|Y_{1:t-1})dX_{t-1} \tag{2}$$

and the state variable can be updated by the posterior density $p(X_t|Y_{1:t})$ inferred from the prior density $p(X_t|Y_{1:t-1})$,

$$p(X_t|Y_{1:t}) = \frac{p(Y_t|X_t)p(X_t|Y_{1:t-1})}{p(Y_t|Y_{1:t-1})} \tag{3}$$

where $Y_{1:t} = \{Y_1, Y_2, \ldots, Y_t\}$ constitutes the complete solution to the sequential estimation problem, and the normalizing constant is

$$p(Y_t|Y_{1:t-1}) = \int p(Y_t|X_t)p(X_t|Y_{1:t-1})dX_t \tag{4}$$

In most cases, the above analytic solution can not be well determined in a direct way. Therefore, the particle filtering by the Monte Carlo simulation is usually taken to approximate the optimal Bayesian estimation recursively.

Let the posterior density function be characterized by N random samples $\left\{X_t^i, \omega_t^i\right\}_{i=1}^N$,

$$p(X_t|Y_{1:t}) \approx \sum_{i=1}^{N} \omega_t^i \delta(X_t - X_t^i) \tag{5}$$

where $\{X_t^i, i = 0, \ldots, N\}$ is a set of support points with the associated weights $\{\omega_t^i, i = 1, \ldots, N\}$.

In the sequential importance sampling, the recursive estimate for the importance weights of the particle i can be derived by

$$\omega_t^i = \omega_{t-1}^i \frac{p(Y_t|X_t^i)p(X_t^i|X_{t-1}^i)}{q(X_t^i|X_{0:t-1}^i, Y_{1:t})} \tag{6}$$

where $q(X_{0:t}|Y_{1:t})$ is an easy to sample and proposal density, $X_{0:t}$ is the historical state variable and $Y_{1:t}$ is the corresponding observation, and the estimated state can be approximated by $\hat{X}_t \approx \sum_{i=1}^N \omega_t^i X_t^i$.

The optimal importance density function will minimize the variance of the true weights, $q(X_t|X_{t-1}^i, Y_t) = p(X_t|X_{t-1}^i, Y_t)$. In practice, it is often convenient to choose the importance density to be the prior,

$$q(X_t|X_{t-1}^i, Y_t) = p(X_t|X_{t-1}^i) \tag{7}$$

Then the importance weight is updated as

$$\omega_t^i = \omega_{t-1}^i p(Y_t|X_t^i) \tag{8}$$

The fish swims as a quite complicated, nonlinear, and non-Gaussian dynamic system in the sea. Theoretically, it is possible to learn the fish behaviors of any complexity if the training database is quite adequate, while in case that the size of samples is in fact far from an optimum [35], we need to offer an efficient and consistent approximation of the dynamic model to estimate the posterior probability density function. The classic particle filtering has been developed as one of the most common and powerful technique for such a system [7,8], while the learning scale, the static reference model, the degeneracy problem, the sample impoverishment, the space dimensionality etc., may still hinder our implementations.

3. The basics of ELM

So far, ELM learning has attracted more and more attention in machine learning since proposed, which announces a novel learning framework that significantly improves generalization performance at surprisingly fast speed, needless of mathematically predetermined internal knowledge [19,20].

Suppose that there are Q arbitrary distinct training samples $\{I_q, O_q\}_{q=1}^Q$, with the input $I_q = [I_{q1}, I_{q2}, \ldots, I_{qd}]' \in R^d$ and the expected output $O_q = [O_{q1}, O_{q2}, \ldots, O_{qk}]' \in R^N$. In general, a standard SLFN can be modeled as the following matrix format,

$$\begin{gathered}\mathbf{H}\boldsymbol{\beta} = \mathbf{O} \\ \mathbf{H}(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{\tilde{Q}}, b_1, \ldots, b_{\tilde{Q}}, \boldsymbol{I}_1, \ldots, \boldsymbol{I}_Q) \\ = \begin{bmatrix} g(\boldsymbol{a}_1 \cdot \boldsymbol{I}_1 + b_1) & \cdots & g(\boldsymbol{a}_{\tilde{Q}} \cdot \boldsymbol{I}_1 + b_{\tilde{Q}}) \\ \vdots & g(\boldsymbol{a}_i \cdot \boldsymbol{I}_q + b_i) & \vdots \\ g(\boldsymbol{a}_1 \cdot \boldsymbol{I}_Q + b_1) & \cdots & g(\boldsymbol{a}_{\tilde{Q}} \cdot \boldsymbol{I}_Q + b_{\tilde{Q}}) \end{bmatrix}_{Q \times \tilde{Q}} \\ \boldsymbol{\beta} = [\boldsymbol{\beta}_1 \quad \cdots \quad \boldsymbol{\beta}_i \quad \cdots \quad \boldsymbol{\beta}_{\tilde{Q}}]'_{k \times \tilde{Q}}, \; \mathbf{O} = [\mathbf{O}_1 \quad \cdots \quad \mathbf{O}_q \quad \cdots \quad \mathbf{O}_Q]'_{k \times Q}\end{gathered} \tag{9}$$

where $\mathbf{H}$ is defined as the hidden layer output matrix, $\tilde{Q}$ is the number of hidden nodes, $g(x)$ stands for the activation function, $\boldsymbol{a}_i = [a_{i1}, a_{i2}, \ldots, a_{id}]'$ is the weight vector connecting the ith hidden node and the input nodes, b_i is the threshold of the ith hidden

Please cite this article as: R. Nian, et al., Extreme learning machine towards dynamic model hypothesis in fish ethology research, Neurocomputing (2013), http://dx.doi.org/10.1016/j.neucom.2013.03.054

node, and $\boldsymbol{\beta}_i = [\beta_{i1}, \beta_{i2}, \ldots, \beta_{ik}]'$ is the weight vector connecting the ith hidden node and the output nodes.

Different from the classical learning algorithm, ELM tends to achieve the least training error and the least norm of output weight together for the better generalization performance by $\boldsymbol{\beta} = \arg\min(||\mathbf{H}\boldsymbol{\beta} - \mathbf{O}||^2, ||\boldsymbol{\beta}||)$, so both the standard optimization method and the minimal norm least square method need to be adopted [19,20]. As long as the hidden layer nodes are enough, SLFN will converge towards any continuous function in ELM with the input weights and hidden layer biases randomly assigned, needless of mathematically predetermined internal knowledge, so that the learning process can be simply considered as a linear system and the output weights can be analytically determined through the Moore–Penrose generalized inverse operation of the hidden layer output matrices.

The basic steps for ELM learning are as follows [19,20].

(1) Randomly assign the input weights $\boldsymbol{a}_i$ and the bias b_i, $i = 1, \ldots, \tilde{Q}$.
(2) Calculate the hidden layer output matrix $\mathbf{H}$.
(3) Calculate the output weights $\boldsymbol{\beta} = \mathbf{H}^{\dagger}\mathbf{O}$, where $\mathbf{H}^{\dagger}$ is the Moore–Penrose generalized inverse of the matrix $\mathbf{H}$.

4. Dynamic models with ELM

4.1. Underwater imaging model

In the fish ethology research, the establishment of the underwater information collection over time depends on a system providing insight of the fish behaviors with great reliability, accuracy and the cost reduction, which strikes a matter of balancing the video quality and the physical limitations (range, resolution, frame rate and compression) while maximizing the memory capacity and battery power.

The underwater images are essentially characterized by the poor visibility, which suffer from the specific properties of the light in the water, such as the limited range, non uniform lighting, low contrast, diminished colors, blur imaging and so on. Moreover, owing to the complexity of the marine environment, the optical properties can often be modified, so the underwater images might have large temporal and spatial variations.

Jaffe–McGlamery model is well-known in the analysis of the underwater image formation [36–38], with the following basic assumptions:

4.1.1. Linear superposition of irradiance

The propagation process at a specific point can be decomposed into three additive linear components,

$$E_t = E_d + E_f + E_b \tag{10}$$

where E_t, E_d, E_f, and E_b are respectively the total irradiance, the direct component, the forward-scattered component and the backscattering. The direct component is the light reflected by the object surface without scattering. The forward scattering is the randomly deviated light reflected by the object on its way to the camera. The backscattering is a significant fraction of the light reflected not by the object but still entered the camera due to the suspended particles in transmission, which causes undesirable differences of contrast and masks the details of the scene.

4.1.2. Attenuation modeling for medium light interaction

The light intensity in the Jaffe–McGlamery model is an exponential decay with distance,

$$L_i(d) = L_{0,i}\exp(-c_i d) \tag{11}$$

where i is the wavelength of light, d is the distance traveling in a liquid, $L_i(d)$ is the light intensity of wavelength i, $L_{0,i}$ is the light intensity of wavelength i at the light source, and c_i is the attenuation coefficient at wavelength i, respectively. The attenuation usually leads to a hazy and poorly contrasted image background.

As a result, the image enhancement is one of the key issues to optimize our understanding [39–41]. In this paper, we present a generic parameter-free enhancement method to make a total abstraction of the image formation process, reduce underwater perturbations, and correct the contrast disparities caused by the attenuation and backscattering, when knowing nothing about the depth, the distance and the water quality. The color space in the image is first converted into the YCbCr space to concentrate only on the luminance channel corresponding to the intensity component. The homomorphic filtering is then adopted to correct non uniform illumination, enhance contrasts and sharpen the edges at the same time. Wavelet decomposition is further introduced to the homomorphic filtering for image denoising. The wavelet base used here is nearly symmetric orthogonal with a bivariate shrinkage exploiting interscale dependency.

4.2. Dynamic model hypothesis

4.2.1. General learning model

In this paper, we try to simply realize the underlying dynamic model of the fish trajectory by means of ELM techniques. We propose to facilitate the entire ELM learning in an ensemble at several levels, including the dynamic state space model, the color distribution model, and the object recognition model. The underlying is to generate multiple versions of tracking cues from different perspective, which when combined, will provide more stable predictions [13,14]. The architecture of one ELM ensemble is shown in Fig. 1.

The first level of the ELM learning starts with an intelligent guess of the current state vector, by feeding the SLFN with a set of previous state vectors over time, so as to memorize the inherent nature reflected from the fish motion. The color distribution, which is robust against non-rigidity, rotation and partial occlusion, is then taken as the fish behavior observation for the ELM architecture at the second level, approximating the color dependence structure and in turn making the feedback to the state vector estimation. The object recognition level is established to have a further understanding to the fish appearances at diverse poses and view angles by implementing ELM in a geometrical topology model [32,42], which will help a lot to make decision in emergency when the tracking fish might be lost.

A collection of the component SLFNs from the individual fish is organized into the hierarchical hybrid ELM ensemble to combine all of the predictions. The combination strategy to the tracking process of the single cues at diverse levels can be considered as the finite mixtures of probability components in a linear fusion strategy. Suppose that the posterior probability density function of the combination in an image is generated from,

$$p(X_t|Y_t, \theta) = \sum_{\ell=1}^{H} \alpha_\ell p_\ell(X_t|Y_t, \theta_\ell). \tag{12}$$

Here $p_\ell(X_t|Y_t, \theta_\ell)$ is the probability density function of each tracking cue at time t, H refers to the total number of the tracking cues, θ_ℓ is the learning parameter during the tracking process, α_ℓ is the mixture proportion with $\sum_{\ell=1}^{H}\alpha_\ell = 1$, to evaluate the importance of the given cue in the observation scene. In case that the tracking cue at each level is of the same important in the dynamic model, the mixture proportion could all set to be equal.

If the minimal time interval between every two concerning frames are sufficiently large, and the images can be assumed

Please cite this article as: R. Nian, et al., Extreme learning machine towards dynamic model hypothesis in fish ethology research, Neurocomputing (2013), http://dx.doi.org/10.1016/j.neucom.2013.03.054

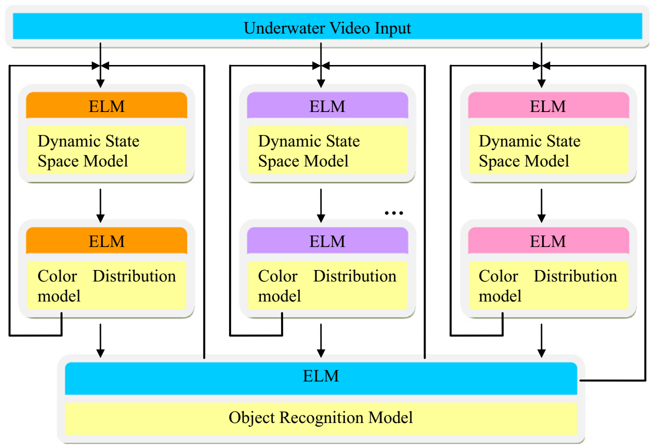


Fig. 1. The architecture of an ELM ensemble.

independently identically distributed (i.i.d.), suppose α_ℓ and θ_ℓ are independent with each other, then the joint probabilistic density function could be expressed as

$$p(\boldsymbol{X}_{t:t+\tau+1}|\boldsymbol{Y}_{t:t+\tau+1},\Theta)=\prod_{i=1}^{\tau+2}\sum_{\ell=1}^{H}\alpha_\ell p_\ell(X_{t+i-1}|Y_{t+i-1},\theta_\ell). \tag{13}$$

Here the parameter set is $\Theta=(\alpha_1,\dots,\alpha_H,\theta_1,\dots,\theta_H)$, and the likelihood function will be taken as

$$L(\Theta|\boldsymbol{X}_{t:t+\tau+1},\boldsymbol{Y}_{t:t+\tau+1})=\sum_{i=1}^{\tau+2}\ln(\sum_{\ell=1}^{H}\alpha_\ell p_\ell(X_{t+i-1}|Y_{t+i-1i},\theta_\ell)). \tag{14}$$

The function form of the probability density function determines the complexity of solving the optimization problem, and it is often analytically not easy to directly analyze the likelihood function above. In simple, the parameter learning in the fish tracking problem could be degraded into the estimation solution of the Gaussian mixture, even when the dynamic models may be non-Gaussian. Some attempts are made to discover the feasibility when realizing the learning adaptively, and correspondingly develop and extend the original ELM algorithms.

4.2.2. Dynamic state space model

Typically, the state vector can be derived from the kinematic and region parameters, so we first define the following dynamic state vector for each single fish,

$$X_t=[x_t,y_t,h_t,w_t,u_t,v_t,s_t]' \tag{15}$$

where (x_t,y_t) and (u_t,v_t) specify the location and motion of the fish, (h_t,w_t) are respectively the image region parameters, $a_t$is the corresponding scale change. The dynamic model is defined as a stochastic differential equation,

$$X_t=AX_{t-1}+U_{t-1} \tag{16}$$

where A describes the deterministic component, and U_{t-1} is the process noise and assumed as a multivariate Gaussian random variable. We currently suppose the motion and the dynamics as the random walk and one first order model is used to represent a region moving with constant velocity and scale change. Expanding this model to other types in a higher order is straightforward.

In the dynamic model, we take a few existing state vectors in the time series as the inputs for SLFN to predict the next state vector with ELM,

$$\mathbf{H}\boldsymbol{\beta}=\boldsymbol{X}$$
$$\mathbf{H}(\boldsymbol{a}_1,\dots,\boldsymbol{a}_{\tilde{Q}},b_1,\dots,b_{\tilde{Q}},\boldsymbol{X}_{0:t-Q},\dots,\boldsymbol{X}_{Q-1:t-1})$$
$$=\begin{bmatrix} g(\boldsymbol{a}_1\cdot\boldsymbol{X}_{0:t-Q}+b_1) & \cdots & g(\boldsymbol{a}_{\tilde{Q}}\cdot\boldsymbol{X}_{0:t-Q}+b_{\tilde{Q}}) \\ \vdots & g(\boldsymbol{a}_i\cdot\boldsymbol{X}_{q:q+t-Q}+b_i) & \vdots \\ g(\boldsymbol{a}_1\cdot\boldsymbol{X}_{Q-1:t-1}+b_1) & \cdots & g(\boldsymbol{a}_{\tilde{Q}}\cdot\boldsymbol{X}_{Q-1:t-1}+b_{\tilde{Q}}) \end{bmatrix}_{Q\times\tilde{Q}}$$
$$\boldsymbol{\beta}=\begin{bmatrix}\boldsymbol{\beta}_1 & \cdots & \boldsymbol{\beta}_i & \cdots & \boldsymbol{\beta}_{\tilde{Q}}\end{bmatrix}'_{k\times\tilde{Q}},\quad \boldsymbol{X}=\begin{bmatrix}\boldsymbol{X}_{t-Q+1} & \cdots & \boldsymbol{X}_{q+t-Q+1} & \cdots & \boldsymbol{X}_t\end{bmatrix}'_{k\times Q} \tag{17}$$

where every input $\boldsymbol{X}_{q:q+t-Q}$ is composed by the state vectors from X_q to X_{q+t-Q}. The time interval $\tau=t-Q$ is a parameter that can be evaluated by the average mutual distance between every pair of inputs. When we feed the SLFN with the input $\boldsymbol{X}_{Q-1:t-1}$, the actual output will be considered as the estimation of the state vectors,

$$\hat{X}_t=\mathbf{H}(\boldsymbol{a},b,\boldsymbol{X}_{t-\tau-1:t-1})\hat{\boldsymbol{\beta}}=\sum_{i=1\tilde{Q}}\beta_i g(a_i\bullet\boldsymbol{X}_{Q-1:t-1}+b_i),$$
$$\hat{\boldsymbol{\beta}}=\mathbf{H}^{\dagger}X,\quad \mathbf{H}^{\dagger}=(\mathbf{H}'H)^{-1}\mathbf{H}' \tag{18}$$

Theoretically, the model selection of the ELM architecture could be evaluated by the generalization error as follows,

$$E=\lim_{Q\to\infty}\sum_{q=1}^{Q}(F(\boldsymbol{X}_{q:q+\tau})-\boldsymbol{X}_{q+\tau+1})^2/Q,\quad q=1,\dots,Q \tag{19}$$

where F is the input–output function of the ELM learning, $\boldsymbol{X}_{q:q+\tau}$ is the τ-dimensional input vector, $F(\boldsymbol{X}_{q:q+\tau})=\hat{\boldsymbol{X}}_{q+\tau+1}$ is the real output of the SLFN corresponding to input, and $\boldsymbol{X}_{q+\tau+1}$ is the expected output.

In practice, leave-one-out crossvalidation is one way to estimate the above procedure, which is basically a special case of k-fold crossvalidation in the case where $k=Q$. The training sets are divided into Q parts, in each one there is exactly one sample that has been left out for testing, and then the estimation of the generalization error becomes,

$$E=\sum_{q=1}^{Q}(F(\boldsymbol{X}_{p:p+\tau},-q)-\boldsymbol{X}_{p+\tau+1})^2/Q,\quad q=1,\dots,Q \tag{20}$$

where $F(\boldsymbol{X}_{p:p+\tau},-q)$ denotes the output of the qth training sets without the qth sample. Here we first start with a large SLFN by the original ELM algorithm, and then rank and eliminate the hidden nodes accordingly. The architecture with the minimum generalization error will be chosen for the state vector estimation. The OP-ELM algorithm with the multi-response sparse regression algorithm (MRSR) and the leave-one-out (LOO) validation could be a good choice to establish a robust and generic dynamic model in SLFN [33,34].

4.2.3. Color distribution model

The color distribution is extracted as the measurement specified by the state vector X_t,

$$p(Y(b)|X_t)=[\sum_{j=1}^{n}L(d(X_t,X_{t,j})/s)\delta(C(x_{t,j},y_{t,j})-b)]/B$$
$$L(d)=\begin{cases}1-d^2 & d<1\\ 0 & \text{otherwise}\end{cases} \tag{21}$$

where (x_t,y_t) is the center of the given region, $(x_{t,j},y_{t,j})$ is the pixel inside the region and n is the number of pixels, $C(x_{t,j},y_{t,j})$ assigns the color to the corresponding bin $b=1,2,\dots,m$, the distribution is discretized into m-bins, δ is the Kronecker delta function, $d(X_t,X_{t,j})$ represents the distance between (x_t,y_t) and $(x_{t,j},y_{t,j})$, $s=\sqrt{h^2+w^2}$ is the parameter to adapt the size of the region, and the normalization factor is $B=\sum_{j=1}^{n}L(d(X_t,X_{t,j})/a)$ so as to ensure the whole probability principle $\sum_{b=1}^{m}p(Y(b)|X_t)=1$. In order to increase the reliability of the color distribution, a weighting function L is employed here, when the boundary pixels belong to the background or get occluded, smaller weights are

Please cite this article as: R. Nian, et al., Extreme learning machine towards dynamic model hypothesis in fish ethology research, Neurocomputing (2013), http://dx.doi.org/10.1016/j.neucom.2013.03.054

assigned to them so that they could be further away from the region center.

Each color distribution in the candidate region will be compared with the reference target model at the origin. The likelihood function here is defined as the Gaussian density,

$$\omega(q(Y_{0,t}|X_t^i)|X_t) = G(D_{0,t}^i|0,\sigma^2), \quad i=1,\ 2,\ldots,\ N \tag{22}$$

where suppose there are N candidate samples around the fish, $q(Y_{0,t}|X_t^i)$ respectively denotes the color distribution of the starting target and the ith candidate in (x_t^i, y_t^i) at time t, $G(D_{0,t}^i|0,\sigma^2)$ is drawn from the Gaussian density with the mean vector 0 and the standard deviation σ, $G(D_{0,t}^i|0,\sigma^2) = \exp(-D_{0,t}^{i\ \ 2}/2\sigma^2)/\sqrt{2\pi}\sigma$, $D_{0,t}^i = D(p_0, q_t^i)$ is the similarity measure between the candidate color distribution $q_t^i = q(Y|X_t^i)$ and the target model $p_0 = p(Y|X_0)$, the weights will be also normalized to ensure the whole probability principle.

The above process adopts one static single target model. Since the samples drawn from the importance density and from the real posterior density will most probably have great deviations, we make use of the ELM algorithm to offer the latest and consistent approximation for fish trajectory tracking problem,

$$\begin{aligned}
&\mathbf{H}\boldsymbol{\beta} = D(p_0, q)\\
&\mathbf{H}(\boldsymbol{a}_1,\ldots,\boldsymbol{a}_{\tilde{Q}}, b_1,\ldots,b_{\tilde{Q}}, \boldsymbol{D}_{0,0:t-Q},\ldots,\boldsymbol{D}_{0,Q-1:t-1})\\
&= \begin{bmatrix} g(\boldsymbol{a}_1\cdot\boldsymbol{D}_{0,0:t-Q}+b_1) & \cdots & g(\boldsymbol{a}_{\tilde{Q}}\cdot\boldsymbol{D}_{0,0:t-Q}+b_{\tilde{Q}})\\ \vdots & g(\boldsymbol{a}_i\cdot\boldsymbol{D}_{0,j:j+t-Q}+b_i) & \vdots\\ g(\boldsymbol{a}_1\cdot\boldsymbol{D}_{0,Q-1:t-1}+b_1) & \cdots & g(\boldsymbol{a}_{\tilde{Q}}\cdot\boldsymbol{D}_{0,Q-1:t-1}+b_{\tilde{Q}}) \end{bmatrix}_{Q\times\tilde{Q}}\\
&\boldsymbol{\beta} = \begin{bmatrix}\boldsymbol{\beta}_1 & \cdots & \boldsymbol{\beta}_i & \cdots & \boldsymbol{\beta}_{\tilde{Q}}\end{bmatrix}'_{k\times\tilde{Q}}, \quad D = \begin{bmatrix}\boldsymbol{D}_{0,t-Q+1} & \cdots & \boldsymbol{D}_{0,j+t-Q+1} & \cdots & \boldsymbol{D}_{0,t}\end{bmatrix}'_{k\times Q}
\end{aligned} \tag{23}$$

where every input vector $\boldsymbol{D}_{0,j:j+t-Q}$ is acquired by the similarity measure from $D_{0,j}$ to $D_{0,j+t-Q}$, and $D_{0,j} = D(p_0, q_j)$ is the similarity measure of the color distribution between the original target model p_0 and the estimated jth mean observation q_j. When we take $\boldsymbol{D}_{0,Q-1:t-1}$ as the input for SLFN to predict the similarity measure $\boldsymbol{D}_{0,t}$, the actual output will be,

$$\begin{aligned}
&\hat{\boldsymbol{D}}_{0,t} = \mathbf{H}(\boldsymbol{a}, b, \boldsymbol{D}_{0,t-\tau-1:t-1})\hat{\boldsymbol{\beta}} = \sum_{i=1}^{\tilde{Q}} \beta_i g(a_i \bullet \boldsymbol{D}_{0,t-\tau-1:t-1} + b_i),\\
&\hat{\boldsymbol{\beta}} = \mathbf{H}^{\dagger} D(p_0, q), \quad \mathbf{H}^{\dagger} = (\mathbf{H}'H)^{-1}\mathbf{H}'
\end{aligned} \tag{24}$$

By the current output $D_{0,t}$ of ELM at time t, an additional weight of the candidate samples will also be considered to retrieve the mutual relationship not with the target model, but with the adjoining sequential changes in the color distribution and the resulting contributions during the tracking process,

$$\omega(q(Y_{Q-1:t-1,t}|X_t^i)|X_t) = G(D'^{i}_{Q-1:t-1,t}|0,\sigma'^2), \quad i=1,\ 2,\ldots,\ N \tag{25}$$

where $q(Y_{Q-1:t-1,t}|X_t^i)$ respectively denotes the color distribution of the state vector sequence from X_{Q-1} to X_{t-1} as well as color distribution of the state vector X_t, in the Gaussian density function $G(D'^{i}_{Q-1:t-1,t}|0,\sigma'^2) = \exp(-D'^{i\ 2}_{Q-1:t-1,t}/2\sigma'^2)/\sqrt{2\pi}\sigma'$, the mean vector and the standard deviation are respectively 0 and σ', $D'^{i}_{Q-1:t-1,t} = D'(D_{0,t}, D_{0,t}^i)$ is another similarity measure function between the prediction $D_{0,t}$ getting from the ELM algorithm and the similarity measure $D_{0,t}^i$ of the color distribution from the ith candidate. The weighting function will then be normalized by $\omega(q(Y_{Q-1:t-1,t}|X_t^i)|X_t) = \omega(q(Y_{Q-1:t-1,t}|X_t^i)|X_t)/\sum_{i=1}^{N}\omega(q(Y_{Q-1:t-1,t}|X_t^i)|X_t)$.

So the complete likelihood for the importance of the candidates can be updated as

$$\begin{aligned}
&\omega(X_t^i|X_t, Y_{0,t}) = k_1\omega(q(Y_{0,t}|X_t^i)|X_t) + k_2\omega(q(Y_{Q-1:t-1,t}|X_t^i)|X_t),\\
&\quad k_1 + k_2 = 1, \quad i=1,\ 2,\ \ldots,\ N
\end{aligned} \tag{26}$$

The sample located around the maximum of the likelihood represents the best match to both the target model and the previous adjoining sequential changes, and the mean state of the sample distribution corresponds well to the maximum and consequently the localization of the fish is more accurate. The Bhattacharyya distance can be one kind of choice for the similarity measure of the color distribution, $D_{t_0,t}^i = \sqrt{1-\rho(p_{t_0}, q_t^i)}$, and $\rho(p_{t_0}, q_t^i) = \sum_{b=1}^{m}\sqrt{p(Y(b)|X_{t_0})q(Y(b)|X_t^i)}$ is the Bhattacharyya coefficient. Fig. 2 is one example of the Bhattacharyya coefficient, the blue dots illustrate the random samples with the previous state, while the yellow dot near the center is the mean state of the sample set, and the green dots represent the sequential importance resampling by the current estimation.

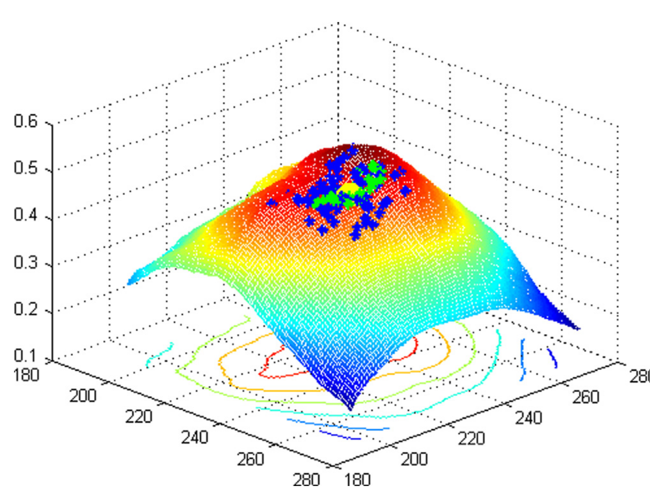


Fig. 2. Bhattacharyya coefficient.

4.2.4. Object recognition model

At the time when the difference between the target model and the estimated candidate reaches the upper limit, it is most likely that the current state vector approaches the image outliers, or the fish is occluded too much, or the images are too noisy, we will not update the target model so as to make sure there is no mistracking when the video surveillance system has lost its tracking.

In this case, an additional fish recognition procedure will get started to keep tracking in emergency. The image region that describes the certain tracking fish at a variety of poses and at multiple view angles are first recorded and stored beforehand. Some pieces of the typical image regions will be selected as the reference regions for the tracking fish. By means of the ELM learning, we take a further step to facilitate the tracking in SLFN by recognizing and locating the certain fish from the existing image regions specified by the candidate samples in aid of one geometrical topology model.

Suppose that there are K fish $\{F_1,\ldots,F_k,\ldots,F_K\}$ in the observation scene, each fish F_i corresponds to M typical regions $\{R_1^k,\ldots,R_m^k,\ldots,R_M^k\}$, and R_m^k denotes the mth region of the certain fish F_k.

In the knowledge of the homologous continuity law, all the pieces of image regions in transition come from the identical fish [32,42–44]. We once put forward one kind of the dipole topology neuron for the ELM learning, to recognize objects in the optimal cognition principle with an extremely huge number of appearances changing in the high-dimensional space [32,42]. The activation function of the dipole topology neuron can be expressed in

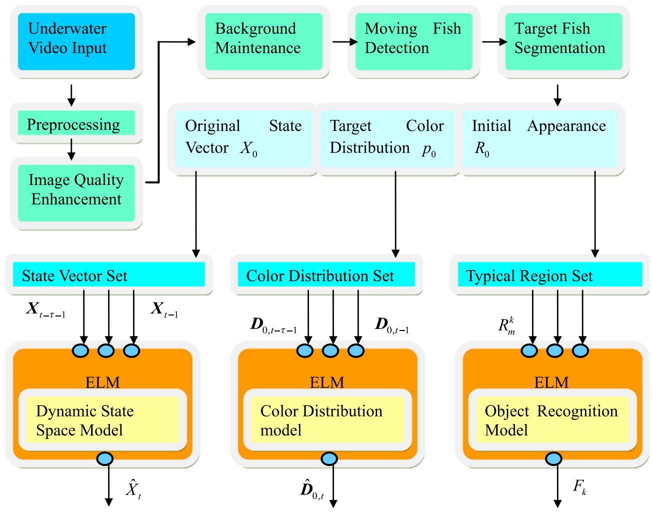


Fig. 3. Initialization.

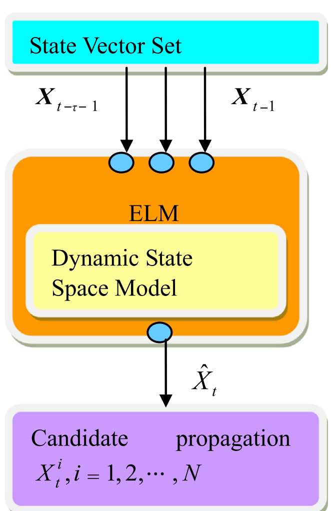


Fig. 4. Propagation.

the following form of the joint probabilistic density,

$$g(R, a_1, a_2, b) = hG(R - f(R, a_1, a_2) \times (a_2 - a_1)|a_1, \Sigma) + b$$
$$f(R, a_1, a_2) = \begin{cases} 0, & \frac{q(R,a_1,a_2)}{d(a_1,a_2)} \le 0 \\ 1, & \frac{q(R,a_1,a_2)}{d(a_1,a_2)} \ge 1 \\ \frac{q(R,a_1,a_2)}{d(a_1,a_2)}, & 0 \le \frac{q(R,a_1,a_2)}{d(a_1,a_2)} \le 1 \end{cases} \tag{27}$$

where $G(R|u, \Sigma)$ follows the Gaussian density with the mean vector u and the covariance matrix Σ, $G(R|u, \Sigma) = \exp\{-\frac{1}{2}(R-\mu)'\Sigma^{-1}(R-\mu)\}/(2\pi)^{n/2}|\Sigma|^{1/2}$, $q(R, a_1, a_2) = (R - a_1)\bullet(a_2 - a_1)/d(a_1, a_2)$, $d(a_1, a_2)$ stands for the similarity measure between two centers a_1 and a_2 of the neuron, b is the bias exerted to the neuron, and h denotes the magnitude of the probabilistic distribution. On a set of typical regions along multiple view angles from the identical fish in an appropriate order with precise position information, a collection of dipole topology neurons could be linked to trace the fish appearances.

Each SLFN consists of three layers: an input layer, a single hidden layer with dipole neurons, and an output layer with the

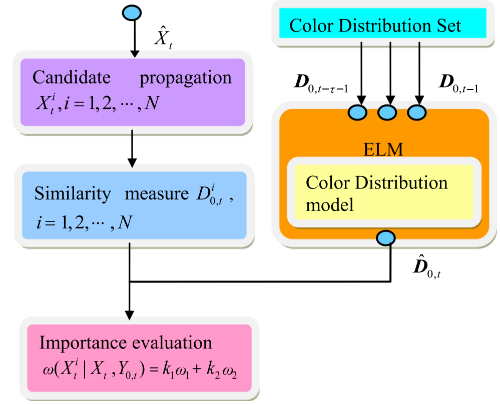


Fig. 5. Evaluation.

linear weights $\boldsymbol{\beta}$, which can be denoted as

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{F}$$
$$\mathbf{H}(\boldsymbol{a}_{11}, \boldsymbol{a}_{21}, \ldots, \boldsymbol{a}_{\tilde{Q}1}, \boldsymbol{a}_{12}, \boldsymbol{a}_{22}, \ldots, \boldsymbol{a}_{\tilde{Q}2}, b_1, b_2 \ldots, b_{\tilde{Q}}, R_1, \ldots, R_Q)$$
$$= \begin{bmatrix} g(R_1, \boldsymbol{a}_{11}, \boldsymbol{a}_{12}, b_1) & g(R_2, \boldsymbol{a}_{11}, \boldsymbol{a}_{12}, b_1) & \cdots & g(R_Q, \boldsymbol{a}_{11}, \boldsymbol{a}_{12}, b_1) \\ g(R_1, \boldsymbol{a}_{21}, \boldsymbol{a}_{22}, b_2) & g(R_2, \boldsymbol{a}_{21}, \boldsymbol{a}_{22}, b_2) & \cdots & g(R_Q, \boldsymbol{a}_{21}, \boldsymbol{a}_{22}, b_2) \\ \vdots & \vdots & \vdots & \vdots \\ g(R_1, \boldsymbol{a}_{\tilde{Q}1}, \boldsymbol{a}_{\tilde{Q}2}, b_{\tilde{Q}}) & g(R_2, \boldsymbol{a}_{\tilde{Q}1}, \boldsymbol{a}_{\tilde{Q}2}, b_{\tilde{Q}}) & \cdots & g(R_Q, \boldsymbol{a}_{\tilde{Q}1}, \boldsymbol{a}_{\tilde{Q}2}, b_{\tilde{Q}}) \\ g(R_2, \boldsymbol{a}_{11}, \boldsymbol{a}_{12}, b_1) & g(R_3, \boldsymbol{a}_{11}, \boldsymbol{a}_{12}, b_1) & \cdots & g(R_1, \boldsymbol{a}_{11}, \boldsymbol{a}_{12}, b_1) \\ g(R_2, \boldsymbol{a}_{21}, \boldsymbol{a}_{22}, b_2) & g(R_3, \boldsymbol{a}_{21}, \boldsymbol{a}_{22}, b_2) & \cdots & g(R_1, \boldsymbol{a}_{21}, \boldsymbol{a}_{22}, b_2) \\ \vdots & \vdots & \vdots & \vdots \\ g(R_2, \boldsymbol{a}_{\tilde{Q}1}, \boldsymbol{a}_{\tilde{Q}2}, b_{\tilde{Q}}) & g(R_3, \boldsymbol{a}_{\tilde{Q}1}, \boldsymbol{a}_{\tilde{Q}2}, b_{\tilde{Q}}) & \cdots & g(R_1, \boldsymbol{a}_{\tilde{Q}1}, \boldsymbol{a}_{\tilde{Q}2}, b_{\tilde{Q}}) \end{bmatrix}_{2\tilde{Q} \times Q}^{'}$$
$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1 & \cdots & \boldsymbol{\beta}_{\tilde{Q}} & \boldsymbol{\beta}_{\tilde{Q}+1} & \cdots & \boldsymbol{\beta}_{2\tilde{Q}} \end{bmatrix}_{K \times 2\tilde{Q}}^{'}, \quad \mathbf{F} = \begin{bmatrix} \boldsymbol{F}_1 & \cdots & \boldsymbol{F}_q & \cdots & \boldsymbol{F}_Q \end{bmatrix}_{K \times Q}^{'} \tag{28}$$

The recognition process is to specify the membership $\boldsymbol{F}_t^i$ of the image region R_t^i around the ith candidate at the time t by the mapping with respect to the ELM learning,

$$\boldsymbol{F}_t^i = \mathbf{H}(\boldsymbol{a}, b, R_t^i)\hat{\boldsymbol{\beta}} = \sum_{j=1}^{\tilde{Q}} \beta_j g(R_t^i, a_{j1}, a_{j2}, b_j)$$
$$+ \sum_{j=\tilde{Q}+1}^{2\tilde{Q}} \beta_j g(R_t^i, a_{(j-\tilde{Q})1}, a_{(j-\tilde{Q})2}, b_{(j-\tilde{Q})})$$
$$\hat{\boldsymbol{\beta}} = \mathbf{H}^{\dagger}\mathbf{F}, \mathbf{H}^{\dagger} = (\mathbf{H}'H)^{-1}\mathbf{H}', \quad i = 1, 2, \ldots, N \tag{29}$$

Here the hidden node number $\tilde{Q}$ is chosen to maintain the principle of the original ELM, which is thereby set as $2\tilde{Q} << Q$.

4.3. ELM learning algorithm

The learning algorithm is composed of the initialization, the propagation, the evaluation, the estimation, the selection and the update, where K and T respectively stand for the number of the fish and the total time period, **For**, **If** and **ELSE** represent the for cycle and if conditional statement. A brief flow chart is shown in Figs. 3–7.

4.3.1. Initialization

Algorithm 1

For $k = 1, 2, \ldots, K$
  Set the initial time interval τ,
  Construct the ELM ensemble for each fish k at distinct levels
  Train each SLFN individually with the prior knowledge getting from a segment of the fish trajectory
  Specify the original state vector X_0 at the time $t = 0$ from the prior, $X_0 \sim p(X_0)$

Get the target model
$p(Y(b)|X_0) = [\sum_{j=1}^{n} L(d(X_0, X_{0,j})/s)\delta(C(x_{0,j}, y_{0,j}) - b)]/B$
Extract the initial image region of the given fish
End

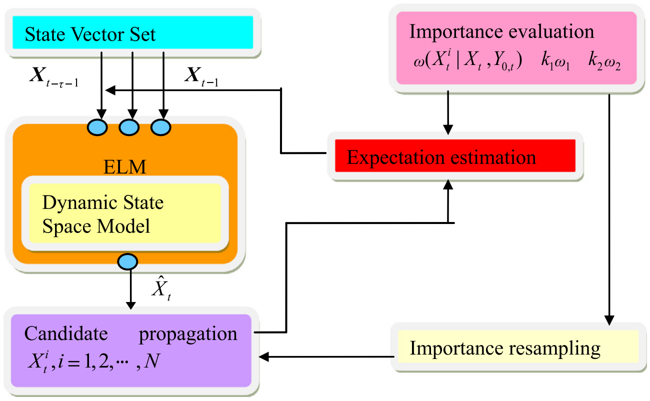


Fig. 6. Estimation and selection.

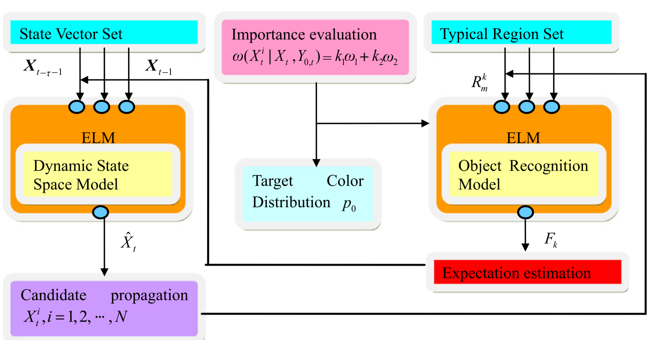


Fig. 7. Update.

4.3.2. Propagation

Algorithm 2
For $t=1, 2, \ldots, T$
For $k=1, 2, \ldots, K$
If $t > \tau$
Predict the current state vector $\hat{X}_t$ with the previous real state in the first level ELM learning
Sample N candidate state vectors X_t^i, $\quad i=1, 2, \ldots, N$ randomly around the $\hat{X}_t$
Else
Assign X_t^i by the previous state vector X_{t-1}
End
End
End

4.3.3. Evaluation

Algorithm 3
For $t=1, 2, \ldots, T$
For $k=1, 2, \ldots, K$
Get the similarity measure prediction $\hat{\boldsymbol{D}}_{0,t}$ of the color distribution between the
original target model p_0 and the estimated current observation by the second level
ELM,
Compute the likelihood to evaluate the importance of the candidate samples based on both the target model and the current observation estimation,
$\omega(X_t^i|X_t, Y_{0,t}) = k_1\omega(q(Y_{0,t}|X_t^i)|X_t) + k_2\omega(q(Y_{t-\tau-1:t-1,t}|X_t^i)|X_t)$, $k_1 + k_2 = 1$, $\quad i=1, 2, \ldots, N$
by normalizing each importance weight, $\omega_t^i = \omega_t^i / \sum_{i=1}^{N} \omega_t^i$.
End
End

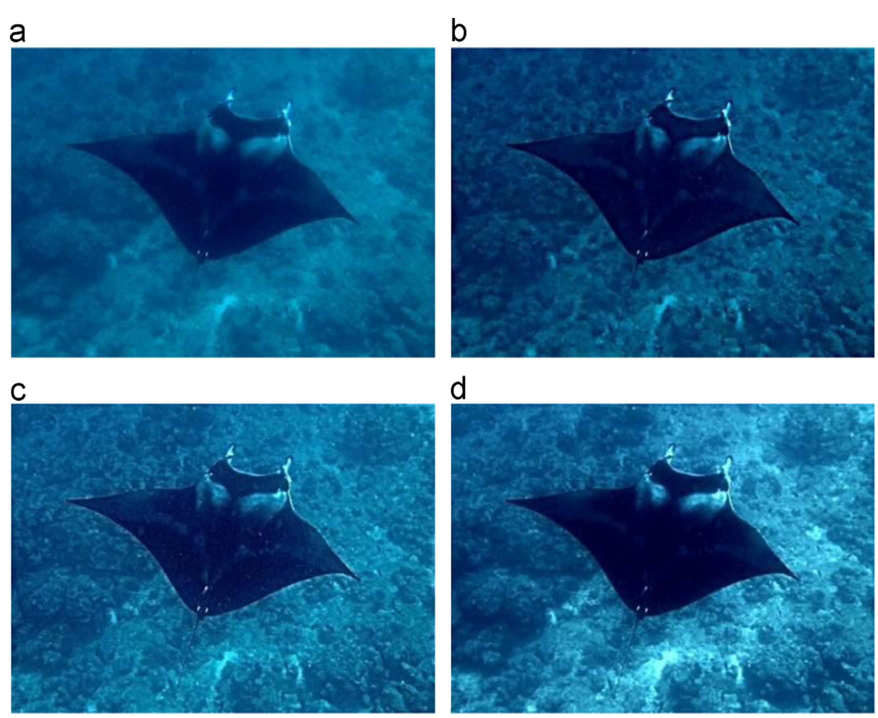


Fig. 8. Image quality enhancement. (a) Original image, (b) wavelet filtering, (c) homomorphic filtering and (d) image enhancement proposed.

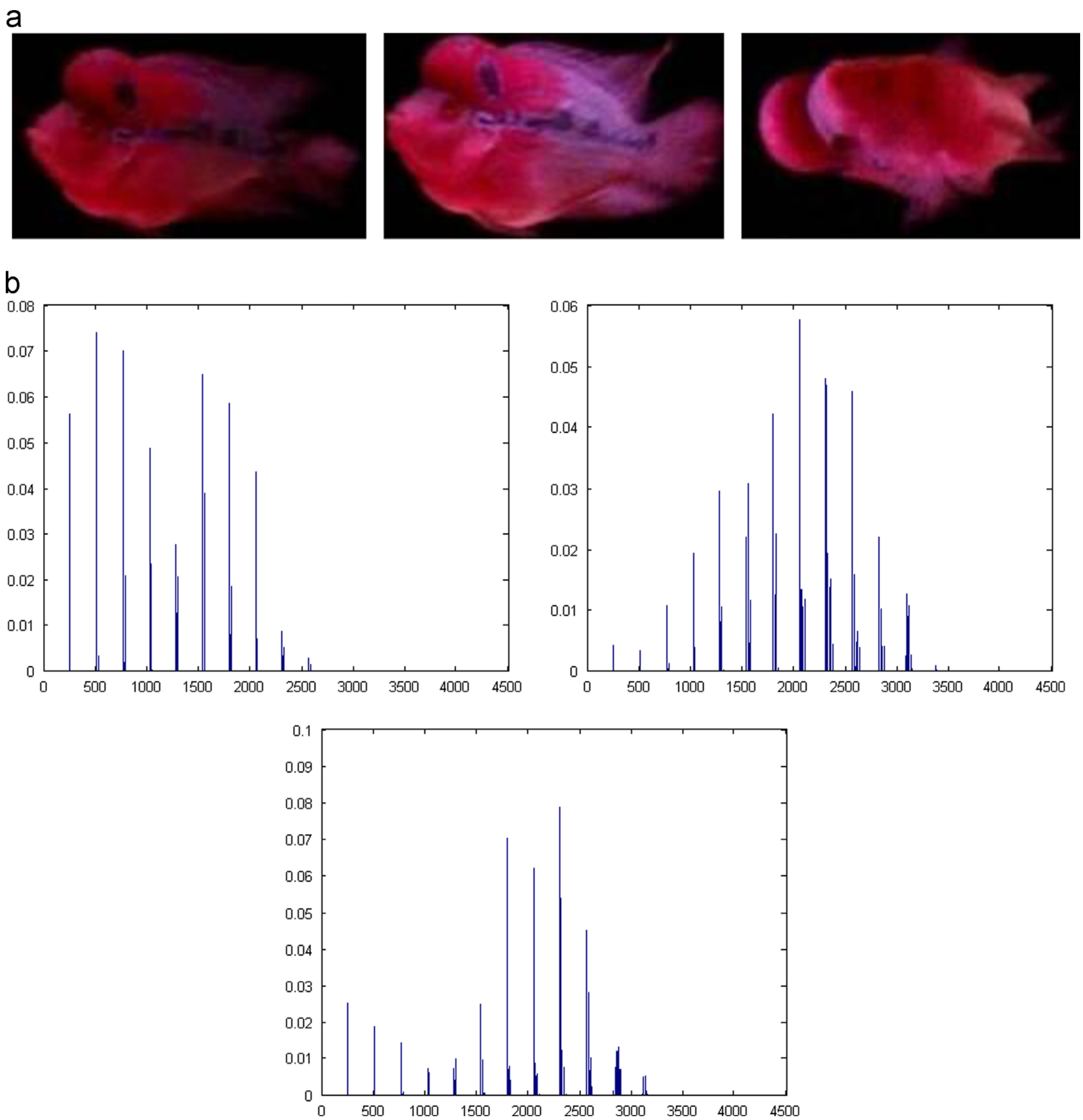


Fig. 9. Color distribution. (a) The image region specified by the state vector and (b) the color distribution of the corresponding image region.

4.3.4. Estimation

Algorithm 4
For $t = 1, 2, \ldots, T$
For $k = 1, 2, \ldots, K$
Estimate the mean state variable
$E[X_t] = \sum_{i=1}^{N} \omega(X_t^i | X_t, Y_{0,t}) X_t^i$, and then set
$\omega(X_t^i | X_t, Y_{0,t}) = 1/N,\ X_t = E[X_t], \quad i = 1, 2, \ldots, N$
Save the current image region of the fish
End
End

4.3.5. Selection

Algorithm 5
For $t = 1, 2, \ldots, T$
For $k = 1, 2, \ldots, K$
Calculate the normalized cumulative probabilities to resample N candidate samples
approximately distributed from $p(X_{t-1} | Y_{t-1})$, each X_t^i will be multiplied or
suppressed according to the importance weight respectively.
End
End

4.3.6. Update

Algorithm 6
For $t = 1, 2, \ldots, T$
For $k = 1, 2, \ldots, K$
Set two thresholds with $0 < l_1,\ l_2 < 1,\ l_1 > l_2$, where l_1 and l_2 are respectively the
thresholds to update the reference and start the emergency

Please cite this article as: R. Nian, et al., Extreme learning machine towards dynamic model hypothesis in fish ethology research, Neurocomputing (2013), http://dx.doi.org/10.1016/j.neucom.2013.03.054

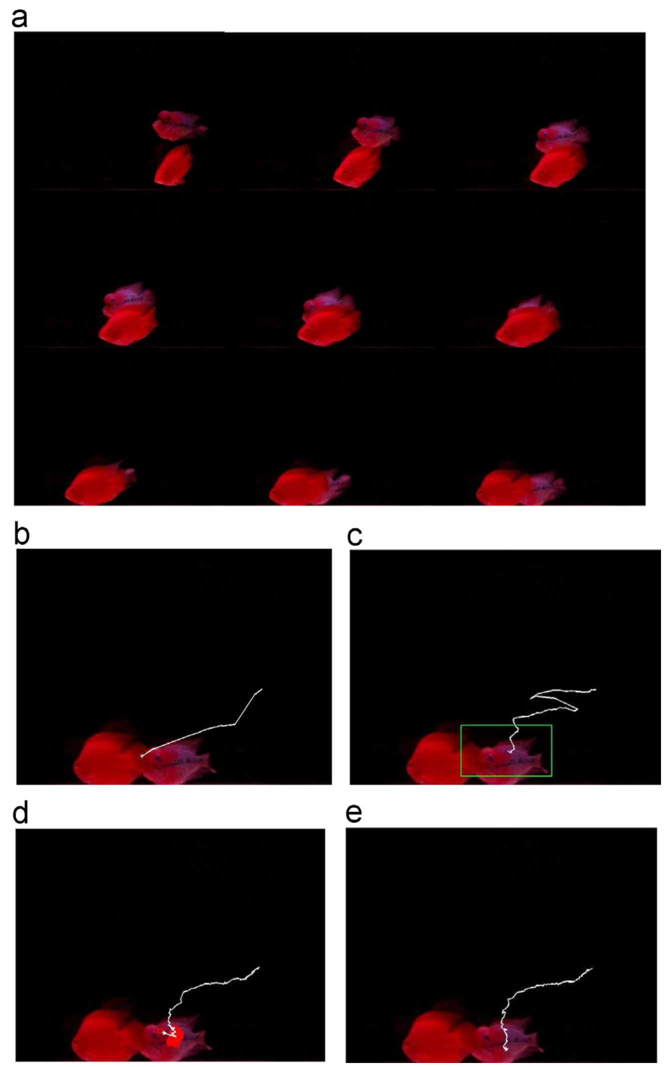


Fig. 10. Fish trajectory tracking for occlusion. (a) Original image sequence in case of occlusion, (b) background subtraction, (c) meanshift (d), particle filtering and (e) ELM tracking proposed.

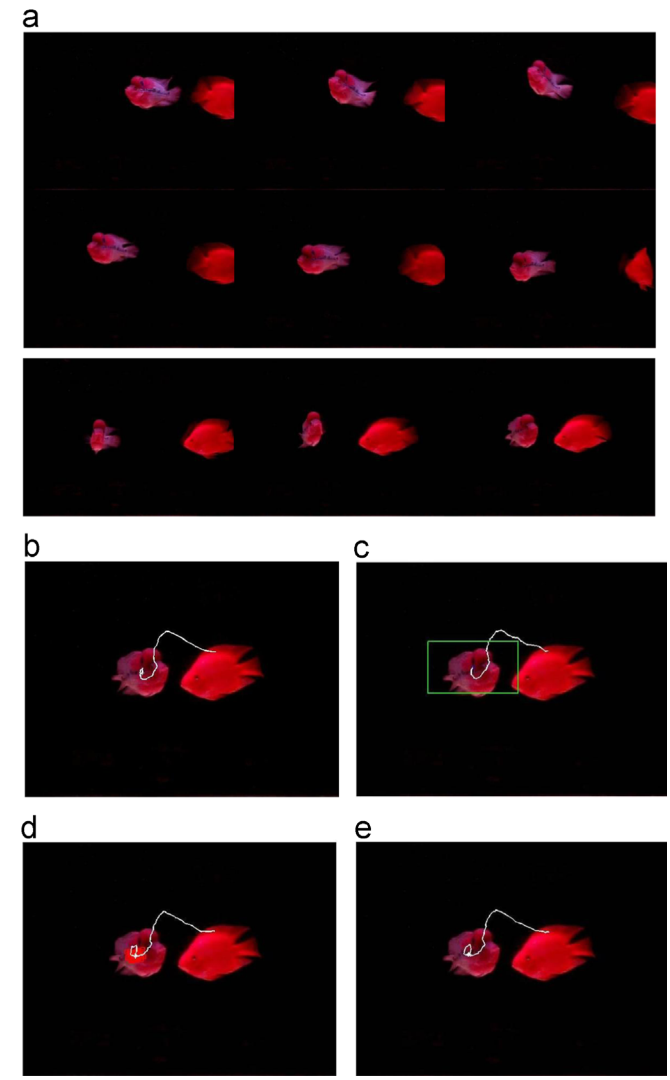


Fig. 11. Fish trajectory tracking for appearance changes. (a) Original image sequence in case of appearance changes, (b) background subtraction, (c) meanshift, (d) particle filtering and (e) ELM tracking proposed.

Set the average weight of the observation over time τ concerning the target
model is
$W(t_0,\tau\Big|X_t)=\frac{1}{\tau+1}\sum_{t=t_0}^{t_0+\tau}\omega(q(Y_{t_0,t}|X_t)|X_t),\quad t_0\ge 0$
If $l_1<W(t_0,\tau|X_t)<1$
Keep the original target model p_0
Else if $l_2<W(t_0,\tau|X_t)<l_1$
Implement the update by
$q(Y(b)|X_t)=(1-\alpha)p(Y(b)|X_0)+\alpha q(Y(b)|X_t)$
Else if $W(t_0,\tau|X_t)<l_2$
Start the emergency recovery and calculate the membership of the given fish $\boldsymbol{F}_t^i$
from the image regions specified by candidate samples with the third level of the
ELM learning
Set the updating threshold during the emergency $0<l_3<1$
Get the vector component $\boldsymbol{F}_t^i(k)$ for the given fish k
Compare and search one with the maximum output from all the candidate samples
for the tracking decision, $i=\arg\max_j \boldsymbol{F}_t^j(k)$
If $\boldsymbol{F}_t^i(k)>l_3$
Assign the state vector X_t^i of the selected candidate sample as the current state
vector X_t
Update the current image region of the fish
Else
Remain in situ and do not update the state vector at this time
End
End
End
End

5. Simulation experiment and result analysis

In the experiments, video recordings were collected by the ROV based surveillance system with the camera mounted. At each observation site, the environmental variables, including the ambient water temperature, current speed at the mooring location, the depth and the direction, as well as the survey-design variables,

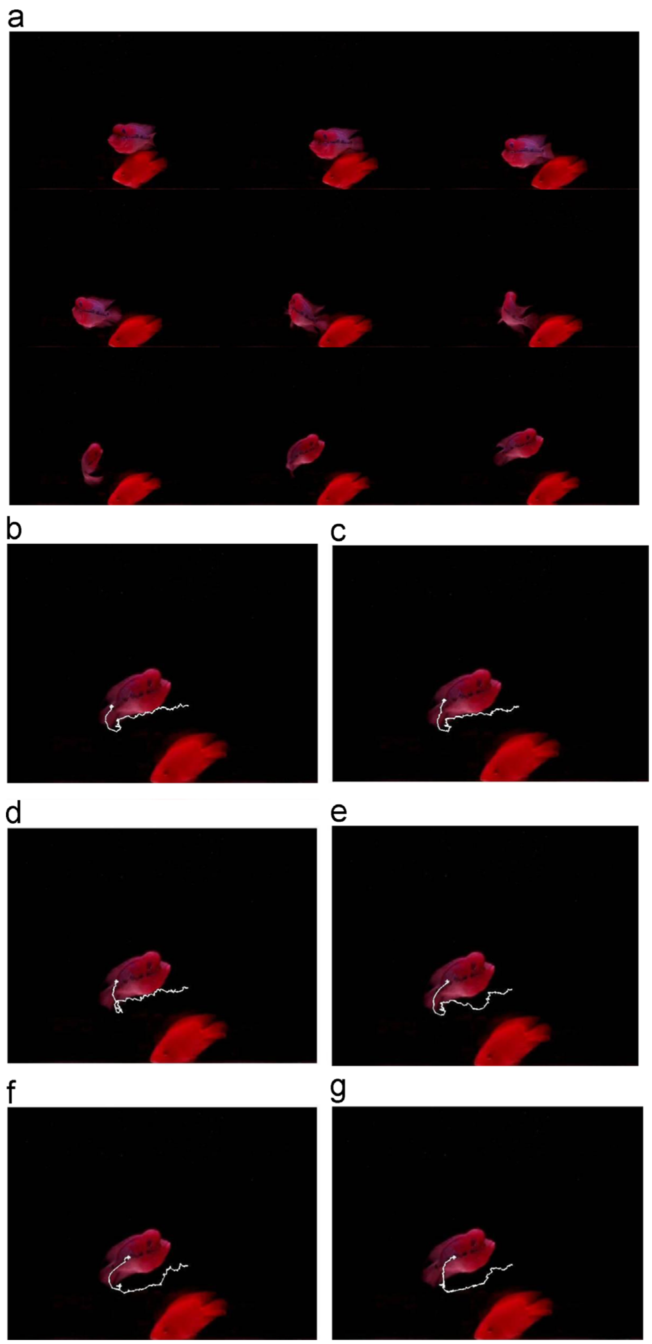


Fig. 12. Fish trajectory tracking by tuning parameters. (a) Original image sequence, (b) $N=20$, $k_1=0.5$, (c) $N=100$, $k_1=0.5$, (d) $N=20$, $k_1=0.7$, (e) $N=100$, $k_1=0.7$, (h) $N=20$, $k_1=1$ and (i) $N=100$, $k_1=1$.

such as the ROV cruising speed and direction, the navigation and positioning, the altitude above the sea floor, ROV distance from the bottom, were recorded simultaneously. The horizontal resolution of the camera is 480 TV lines and the scanning is 625Line/50 Hz PAL. Trajectory tracking was conducted on the fish image sequences in collection. The image size is 480×640 pixels. All the simulation experiments have been run on the same x86_64 Windows machine with at least 4 GB of memory and 2+GHz processor. The execution environment is under MATLAB 7.0 on the basis of the OP-ELM.

Some preprocessing was then done to decrease noise or fulfill feature extraction in advance before formally fish trajectory tracking. One underwater enhancement method we present was adopted here to get better image quality for the fish ethology research. Fig. 8 lists the resulting enhancements for one example fish image by the wavelet filtering, the homomorphic filtering, as well as the proposed method. The state vectors, the color distribution, and some typical regions for the given fish were stored into a working memory and accumulated over time, which are indispensable to build the initial ELM architecture at diverse levels in our simulation. Fig. 9 shows some examples of the image region and the corresponding color distribution for one fish respectively.

Simulations were taken on 100 video clips selected from the collection with the observation objects in the scene to perform the tracking. Figs. 10 and 11 are respectively fish trajectory tracking results for the occlusion and appearance changes that exist commonly during surveillance.

The background subtraction adopted here would find any minor change between the current image and the background, which had much more dependences to the preprocessing techniques, especially the denoising and enhancement of the collected video. When there were fish approaching each other, it would not come into effect at that time.

The mean shift method was efficient to eliminate the brute force to trace fish trajectory in real time during the whole sequence, which maximized the similarity measure iteratively by comparing the candidate image region with the window around the hypothesized fish location until convergence was achieved. Since the search chose the most probable image region as a whole every time, the portion of the given fish needed to be inside the image region upon initialization, when there were fish overlapping or

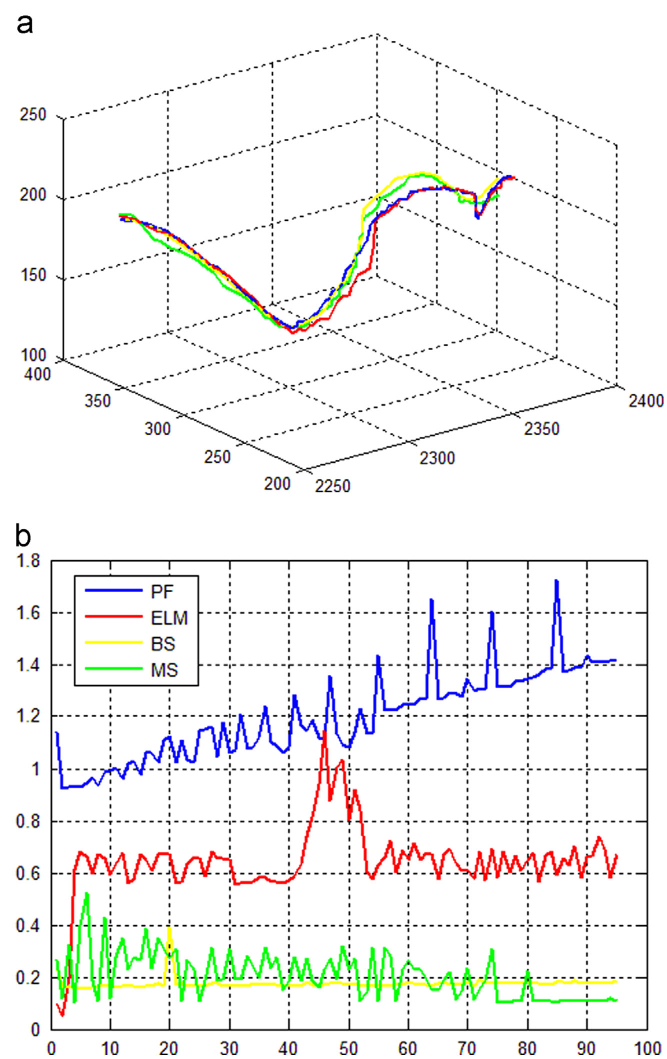


Fig. 13. Tracking results. (a) Tracking curves and (b) time periods.

Please cite this article as: R. Nian, et al., Extreme learning machine towards dynamic model hypothesis in fish ethology research, Neurocomputing (2013), http://dx.doi.org/10.1016/j.neucom.2013.03.054

Table 1
Tracking performances

Performances	BS	MS	PF *N*			ELM *N*		
			20	60	100	20	60	100
Time period	0.1747	0.2043	0.3261	0.8735	1.1984	0.2305	0.4631	0.6443
MSE	11.7016	7.0152	7.2758	6.4534	6.6514	7.1726	6.0896	7.6508

moving at too fast speed in one scene, it might result in the lost of the tracking fish. In case of fish rotation around the nearby location, the mean shift usually could not change the tracking window location too much, and the center of the tracking window sometime might not describe the real trajectory so exactly.

Particle filtering was a powerful and reliable tool for fish trajectory tracking during surveillance. Fig. 10(d) is the tracking result of the particle filtering by the current color distribution purely compared to the target model. Fig. 11(d) is also the particle filtering result for another example image sequence with the current color distribution and the target model concerned. Since the generic method maintained the single static motion model and could not offer a dynamic and consistent approximation of state variables for target tracking, the tracking would have more deviation to the given fish when the appearances changed sharply and could not overcome the cumulation of estimation errors. The degeneracy problem also occurred in the particle filtering when all but one particle had significant weight.

The ELM algorithm proposed in this paper made the decision by several diverse cues extracted from the dynamic model, and tried to perform a robust and accurate approximation to the state variables. The posterior distribution error of each candidate state vector would not have the impact on the performance too much when tracking by a series of previous observations and the prediction based on observations, and could approximate the real fish trajectory effectively. With the guidance of ELM learning, each candidate state vector could estimate the probability distribution more effectively and keep the multimodality at the same time.

Furthermore, the tracking results by tuning some parameters are shown in Fig. 12, where N is the number of the candidate state vectors and k_1 is the impact weight of the original target model. We also made a comparison in Fig. 13 on the tracking curves and time periods of different tracking techniques for one example video clip when setting $N = 100$, i.e., the background subtraction (BS), the mean shift (MS), the particle filtering (PF) and the proposed ELM algorithm. Table 1 lists the average statistic performances for the above four tracking methods, including the time periods as well as the mean squared error (MSE) between the tracking results and real fish trajectory. In the simulation experiments, it was not necessary for the background subtraction to initialize and train the tracking process beforehand, while for the three others, the original region of each fish was required to specify and define in advance, and for the ELM based method, we also need to first construct and train the SLFNs. It is shown here that the tracking performances with ELM were comparable to the particle filtering. Although the computation complexity of the tracking increased when introducing the ELM learning, at each time instant there was only one test process and the time was irrelative with the candidate state vectors, and the execution time in our method is less than particle filtering.

6. Conclusions

In this paper, we have focused on a scheme of the dynamic model approximation for the fish ethology research based on ELM learning. The fish trajectory has been tracked by the multiple previous cues and current predictions getting from the hierarchical hybrid ELM ensemble with the dynamic state space model, the color distribution model, and the object recognition model. The relevant mathematical criterion have been developed to perform the candidate propagation, the importance evaluation, the resampling and selection, the target model update, and the potential appearance recognition. The simulation results have shown the effectiveness and feasibility of the proposed approach, which is comparable to the classical tracking algorithms, even if the occurrence of the occlusion, the deformation, or missing.

Acknowledgments

This work was fully supported by the Natural Science Foundation of P.R. China (31202036), the National Science & Technology Pillar Program of P.R. China (2012BAD28B05), the Natural Science Foundation of P. R. China (41176076), and the Science & Technology Key Program of Shandong Province, P. R. China (2013 GHY11507).

References

[1] D.I. Mostofsky, The Behavior of Fish and Other Aquatic Animals, Academic Press, New York–San Francisco–London, 1978.
[2] M.H.A. Keenleyside, Diversity and adaptation in fish behaviour, in: H. Caspers (Ed.), Zoophysiology and Ecology. Springer-Verlag, 1979.
[3] P. Fréon, L. Dagorn, Review of fish associative behaviour: toward a generalization of the meeting point hypothesis, Rev. Fish Biol. Fish. 10 (2000) 183–207.
[4] M. Lowry, H. Folpp, M. Gregson, R. Mckenzie, A comparison of methods for estimating fish assemblages associated with estuarine artificial reefs, Braz. J. Oceanogr. 59 (2011). (no.spe1, São Paulo).
[5] A. Yilmaz, O. Javed, M. Shah, Object tracking: a survey, ACM Comput. Surv. 38 (2006) 4, http://dx.doi.org/10.1145/1177352.1177355. (Article 13).
[6] D. Comaniciu, V. Ramesh, P. Meer, Kernel-based object tracking, IEEE Trans. Pattern Anal. Mach. Intell. 25 (5) (2003) 564–577.
[7] J. Czyz, B. Ristic, B. Macq, A particle filter for joint detection and tracking of color objects, Image Vision Comput. 25 (8) (2007) 1271–1281.
[8] K. Nummiaro, E. Koller-Meier, L. Van Gool, A color-based particle filter, Image Vision Comput. 21 (1) (2003) 99–110.
[9] J.Q. Smith, Decision Analysis: A Bayesian Approach, Chapman & Hall, Ltd., London, UK, 1987.
[10] J.O. Berger, Statistical Decision Theory and Bayesian Analysis, Springer, New York, 1993.
[11] P. Fearnhead, Sequential Monte Carlo Methods in Filter Theory, Merton College, Oxford, 1998.
[12] S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice-Hall New Jersey, 1999.
[13] L.K. Hansen, Neural network ensembles, IEEE Trans. Pattern Anal. Mach. Intell. 12 (10) (1990) 993–1001.
[14] A. Krogh, J. Vedelsby, Neural network ensembles, cross validation, and active learning, in: G. Tesauro, D. Touretzky, T. Leen (Eds.), Advances in Neural Information Processing, Systems 7, Denver, CO, MIT Press, Cambridge, MA, 1995, pp. 231–238.
[15] M. Verleysen, J.L. Voz, P. Thissen, J.D. Legat, A statistical neural network for high-dimensional vector classification, in: Proceedings of the IEEE International Conference on Neural Networks, ICNN'95, Perth, pp. 990–994, 1995.
[16] S.J. Wang, B.N. Wang, Analysis and theory of high-dimension spatial geometry for Artificial Neural Networks, Acta Electron. Sin. 30 (1) (2002) 1–4.
[17] G.B. Huang, Learning capability and storage capacity of two hidden-layer feedforward networks, IEEE Trans. Neural Networks 14 (2) (2003) 274–281.
[18] A. Lendasse, V. Wertz, M. Verleysen, Model selection with cross-validations and bootstraps – application to time series prediction with RBFN models, in: Proceedings of the ICANN 2003, Joint International Conference on Artificial Neural Networks, Istanbul, Turkey, vol. 2714, pp. 573–580, 2003.
[19] G.B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (2006) 489–501.

Please cite this article as: R. Nian, et al., Extreme learning machine towards dynamic model hypothesis in fish ethology research, Neurocomputing (2013), http://dx.doi.org/10.1016/j.neucom.2013.03.054

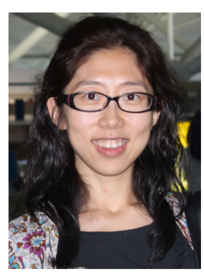
[20] G.B. Huang, D.H. Wang, Y. Lan, Extreme learning machines: a survey, Int. J. Mach. Learn. Cyber. 2 (2011) 107–122.
[21] G.B. Huang, X. Ding, H. Zhou, Optimization method based extreme learning machine for classification, Neurocomputing 74 (2010) 155–163.
[22] G.B. Huang, L. Chen, C.K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Trans. Neural Networks 17 (4) (2006) 879–892.
[23] G.B. Huang, L. Chen, Convex incremental extreme learning machine, Neurocomputing 70 (2007) 3056–3062.
[24] G.B. Huang, L. Chen, Enhanced random search based incremental extreme learning machine, Neurocomputing 71 (2008) 3460–3468.
[25] G.B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multi-class classification, IEEE Trans. Syst. Man. Cyber. 42 (2) (2012) 513–529.
[26] Z.H. Man, K. Lee, D.H. Wang, Z.W. Cao, C.Y. Miao, A new robust training algorithm for a class of single-hidden layer feedforward neural networks, Neurocomputing 74 (2011) 2491–2501.
[27] Z.H. Man, K. Lee, D.H. Wang, Z.W. Cao, C.Y. Miao, A modified ELM algorithm for single-hidden layer feedforward neural networks with linear nodes, in: Proceedings of the 2011 6th IEEE Conference on Industrial Electronics and Applications, pp. 2524–2529, 2011.
[28] B. Frénay, M. Verleysen, Using SVMs with randomized feature spaces: an extreme learning approach, in: Proceedings of the 18th European Symposium on Artificial Neural Networks (ESANN), Bruges, Belgium, 28–30 Apr 2010, pp. 315–320, 2010.
[29] B. Frénay, M. Verleysen, Parameter-insensitive kernel in extreme learning for non-linear support vector regression, Neurocomputing 74 (16) (2011) 2526–2531.
[30] M. van Heeswijk, Y. Miche, T. Lindh-Knuutila, P.A. Hilbers, T. Honkela, E. Oja, A. Lendasse, Adaptive ensemble models of extreme learning machines for time series prediction, Lect. Notes Comput. Sci. 5769 (2009) 305–314.
[31] M. van Heeswijk, Y. Miche, E. Oja, A. Lendasse, Gpu accelerated and parallelized ELM ensembles for large-scale regression, Neurocomputing 74 (2011) 2430–2437.
[32] R. Nian, B. He, A. Lendasse, 3D object recognition based on a geometrical topology model and extreme learning machine. Extreme learning machine's: theory and application, Neural Comput. Appl. 22 (3–4) (2012) 427–433, http://dx.doi.org/10.1007/s00521-012-0892-7.
[33] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, A. Lendasse, OP-ELM: optimally pruned extreme learning machine, IEEE Trans. Neural Networks 21 (2010) 158–162.
[34] Y. Miche, M. van Heeswijk, P. Bas, O. Simula, A. Lendasse, TROP-ELM: a double-regularized ELM using LARS and Tikhonov regularization, Neurocomputing 74 (2011) 2413–2421.
[35] M. Verleysen, Learning high-dimensional data, in: S. Ablameyko (Ed.), Limitations and Future Trends in Neural Computation, IOS Press, Amsterdam, 2003, pp. 141–162.
[36] J.S. Jaffe, Computer modeling and the design of optimal underwater imaging systems, IEEE J. Oceanic Eng. 15 (2) (1990) 101–111.
[37] J.S. Jaffe, K.D. Moore, J. McLean, M.P. Strand, Underwater optical imaging: status and prospects, Oceanography 14 (3) (2001) 66–76.
[38] R. Schettini, S. Corchs, Underwater image processing: state of the art of restoration and image enhancement methods, EURASIP J. Adv. Signal Process. 746025 (2010).
[39] A. Arnold-Bos, J.P. Malkasse, G. Kerven, A preprocessing framework for automatic underwater images denoising, in: Proceedings of the European Conference on Propagation and Systems, Brest, France, 2005.
[40] S. Bazeille, I. Quidu, L. Jaulin, J.P. Malkasse, Automatic underwater image pre-processing, in: Proceedings of CMM'06, Brest, France, 2006.
[41] K. Iqbal, R. Abdul Salam, M.A. Osman, A.Zawawi Talib, Underwater image enhancement using an integrated colour model, IAENG Int. J. Comput. Sci. 32 (2) (2007) 239–244.
[42] G.R. Ji, R. Nian, S.M. Yang, L.J. Zhou, C. Feng, Cellular recognition for species of phytoplankton via statistical spatial analysis, in: Proceedings of the Lecture Notes in Control and Information Sciences, Intelligent Computing in Signal Processing and Pattern Recognition, ICIC, pp. 761–766, 2006.
[43] J.J. Koenderink, A.J. van Doorn, The singularities of the visual mapping, Biol. Cyber. 24 (1976) 51–59.
[44] S.J. Wang, Biomimetics Pattern Recognition, INNS, ENNS, JNNS Newletters Elseviers, 2003.

Rui Nian is an assistant professor in the Information Science and Engineering College at Ocean University of China (OUC). Her major is ocean information detection and processing. She received her Bachelor degree in signal and information processing from OUC, and got her Doctor degree both from OUC and Université catholique de Louvain (UCL). Her primary research interests include pattern recognition, computer vision, image processing, cognitive science, machine learning, and high dimensional space analysis. She is also jointly appointed as the visiting professor in the ICTEAM at UCL and the research scientist in the the Research and Development Center, Haier Group currently.

Bo HE received his master degree and Ph.D. degree from the Harbin Institute of Technology, China, in 1996 and 1999, respectively. From 2000 to 2003, he has been with Nanyang Technological University (Singapore) as a Post-Doctoral Fellow, where he worked on mobile robots, unmanned vehicles, research works included precise navigation, control and communication. In 2004, he joined the Ocean University of China, and now he is a full-time professor and deputy head of Department of Electronics Engineering in Information Science and Engineering College. In the past 10 years, he has been working on underwater robot, especially on positioning and navigation of AUVs. His current research interests include AUV design and applications, AUV SLAM, AUV control, and machine learning.

Bing Zheng received the B.A. degree from the Ocean University of China in Electronics and information System in 1991, the M.S. degree in oceanophysics from the Ocean University of China in 1995, and the Ph.D. degree in computer application from the Ocean University of China in 2013. He is currently a professor and head of the electronics department in Information Science and Engineering College, Ocean University of China. His research interests are in the areas of underwater vision detection.

Mark van Heeswijk has been working as an exchange student in both the EIML (Environmental and Industrial Machine Learning, previously TSPCi) Group and Computational Cognitive Systems Group on his Master's Thesis on “Adaptive Ensemble Models of Extreme Learning Machines for Time Series Prediction”, which he completed in August 2009. Since September 2009, he started as a Ph.D. student in the EIML Group, ICS Department, Aalto University School of Science. His main research interests include high-performance computing, scalable machine learning methods, ensemble models and neural networks like extreme learning machines and deep belief networks.



Qi Yu her first Master degree from Harbin Institute of Technology (HIT) in 2005, China, majoring telecommunication. After that she came to Finland and got her second Master degree about approximation problems in Finance in Aalto University School of Science and Technology (previously Helsinki University of Technology) in Finland, 2007. Now, she is doing her doctoral studies in Environmental and Industrial Machine Learning Group in Aalto University, her research interests include machine learning methods, missing data problem, feature selection, ensemble modeling, neural network and most particularly, for bankruptcy prediction.

Yoan Miche received an Engineer's Degree from Instituti National Polytechnique de Grenoble (INPG, France), and more specifically from TELECOM, INPG, on September 2006. He also graduated with a Master's Degree in Signal, Image, Speech and Telecom from ENSERG, INPG, at the same time. He is currently working in the ICS lab at Aalto University as a postdoctoral researcher, after obtaining a D.Sc. from INPG (France) and Aalto University (Finland) in 2010. His main research interests are currently in Machine Learning for classification/regression tasks in Internet Security.

Amaury Lendasse got his first Master Degree in Mechanical Engineering and his second Master Degree in Control at Université Catholique de Louvain, Belgium in 1997. He got his Ph.D. degree in 2003 in the same university under the supervision of Prof. Vincent Wertz and Michel Verleysen. He is now a docent and senior researcher in the Department of Information and Computer Science, Aalto University. His research interests cover machine learning, time-series prediction, environmental modeling, industrial applications, information security, variable selection, etc.