Neurocomputing 104 (2013) 170–179

Contents lists available at SciVerse ScienceDirect

Neurocomputing

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

A biologically inspired solution to simultaneous localization and consistent mapping in dynamic environments

Yangming Li [a,*], Shuai Li [b], Yunjian Ge [a]

[a] Robot Sensor and Human–Machine Interaction Lab., Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei, Anhui 230031, China
[b] Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA

ARTICLE INFO

Article history:
Received 3 June 2012
Received in revised form
17 August 2012
Accepted 24 October 2012
Communicated by Dr. Tao Mei
Available online 19 November 2012

Keywords:
Shunting short term memory model
Scan matching
Simultaneous localization and consistent mapping
Dynamic environment
Biological inspiration

ABSTRACT

Simultaneous localization and consistent mapping in dynamic environments is a fundamental and unsolved problem in the mobile robotics community. Most of the algorithms for this problem heavily rely on discriminating dynamic objects from static objects. Because these recursive filters based discrimination algorithms always have lag before the model selection parameters converge to the steady states, they have a period of time that the filter could identify a dynamic target as static or vice versa. Mis-classifications decrease precision and consistence, and induce filter divergence.

A brain interacts with dynamic environments. The biological basis of this adaptability is provided by the connectivity and the dynamic properties of neurons. Biologically inspired by the adaptability, the paper proposes a shunting STM (Short Term Memory) based method to solve the simultaneous localization and consistent mapping problem, especially in dynamic environments. The proposed method utilizes a shunting STM neural network to represent environments and to probabilistically reflect the probability of existence of an object; it adapts a scan matching scheme to localize robot based on the map representation. Dynamic properties of the neural network are used to reflect environmental changes, therefore, the proposed method does not require explicit discrimination of objects. As a result, the proposed method does not have the lag of convergence, and it has high utilization ratio of observation information. Theoretical analyses in the paper show the proposed method has Lyapunov stability and its computational complexity does not depend on the size of the environment. The paper compares the proposed method with the classification based Extend Kalman Filter on a classical outdoor dataset, in simulated environments and in real indoor environments. The results show the proposed method outperforms the classification based EKF on precision and consistence in both static environments and dynamic environments.

1. Introduction

Solutions to the Simultaneous Localization and Mapping (SLAM) problem enable a mobile robot that is placed at an unknown location in an unknown environment to incrementally build a consistent map of the environment while simultaneously determining its location within the map. The SLAM problem is appealing to lots of researchers, because it is widely regarded as the foundation of truly autonomous robotics [1,2]. The research on the SLAM problem is one of the notable successes of the robotics community over the past decade. Unfortunately, the more practical version of the problem, SLAM in dynamic environments, still remains as a challenge and desires a lot of efforts [1–3].

Real world are not static. They contain both moving objects, such as cars and people, and temporary structures that do not have stable positions, such as doors and chairs. Traditionally, the SLAM problem in dynamic environments is treated as a model selection problem [1,2,4]. The mobile robot constantly makes decision on whether or not a landmark is stationary. More specifically, some of the algorithms divide the problem into two separate parts: a standard SLAM problem and a multi-target tracking problem [5,6]; some of them eliminate dynamic objects and degenerate the problem to the static SLAM problem [7,8]; some of them exclude dynamic objects in the data association process [10,4,9]; some of them avoid to extract features from dynamic objects [11,12]. All these classification based algorithms share one common disadvantage: they are error-prone, because there is always a lag between the requirement on making decision and accumulation of enough proof to make a decision [4,1].

When the brain interacts with the environment it constantly adapts by representing the environment in a form that is called an internal model; the biological basis for internal models is provided

* Corresponding author.
E-mail address: ymli@iim.ac.cn (Y. Li).

0925-2312/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2012.10.011

by the connectivity and the dynamical properties of neurons [13,14]. The shunting STM (Short Term Memory) model describes how individuals adapt their memory in real-time to complex and changing environments [15]. The model is derived from the additive STM model through bounding activities of neurons [15,16]. This family of neural networks provides a model for understanding human memory and is widely used for pattern recognition and nonlinear learning [17,18]. Inspired by the adaptability of memory, the paper proposes to avoid explicit object classification through utilizing dynamic properties of neurons. The key of the proposed method is to simulate the process of which a brain memorizes environments and to dynamically update memory. Fortunately, the shunting STM network model is guaranteed to converge to a local minimum, but convergence to one of the stored patterns is not guaranteed [15]. Therefore, in our proposed method, we can use neurons to store possibilities of existence of both static objects and dynamic objects. This probabilistic map representation can be easily combined with scan matching based localization to solve the SLAM problem in both static and dynamic environments.

It has a long history of solving robotics problems with biological algorithms [19–22]. Even in the SLAM community, the computational models of the rodent hippocampus was introduced to solve the SLAM problem in large environments [21]. This algorithm adopts a structure being similar to the rodent hippocampus to represent robot poses, and uses lightweight vision systems to recognize seen scenarios. The algorithm is for large and static environments. Shunting STM models were also used to represent maps before [22,23]. Although similar network models are adopted to represent maps, meaning of neuron activities are very different. In the previous works, activities of neurons denote attractions from a target to a mobile robot (for path planning); in our method, neuron activities are confidence on existence of obstacles.

The central contributions of this paper are as follows:

- We propose a biological inspired solution to the SLAM problem, especially in dynamic environments; the proposed method does not require to discriminate dynamic objects from static objects, and it has solid probabilistic foundation.
- We show the proposed method has Lyapunov stability, through theoretical analysis.
- We show the proposed method has constant computational complexity and small space complexity, through theoretical analysis.
- We show the proposed method has high localization precision and is capable of mapping consistently in both static and dynamic environments, through both dataset verifications, software simulation and experiments in real environments.

The paper is organized as follows. Section 2 presents some preliminaries on both the definition of the SLAM problem and shunting STM models. Section 3 explains the proposed method in detail, including: shunting STM equation based map representation, theoretical analysis of stability of the neural network, scan matching based localization and analyses of the computational complexity and the space complexity of the proposed method. In Section 4, the proposed method is compared with the classification based EKF algorithm in a classical dataset, simulated environments and real environments. Conclusions are drawn in Section 5.

2. Preliminaries

2.1. SLAM Problem statement

The SLAM problem is commonly modeled as a Dynamic Bayesian Network [2], as shown in Fig. 1. In the figure, circles denote random variables. While a robot moves, the current position of the robot, x_t, depends on both the previous location, x_{t-1}, and the transformation, u_t; observations, z_t, depends on the current robot pose and the observable landmarks, m. Dependencies are indicated by arrows in Fig. 1.

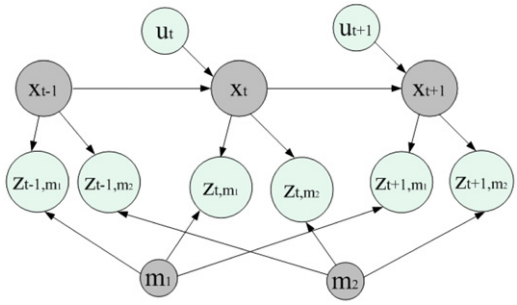


Fig. 1. Modeling SLAM as a dynamic Bayesian network. Arrows indicate the dependence among random variables; light blue circles indicate observable random variables; dark gray circles indicate random variables we need to estimate. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The SLAM problem is mathematically defined as [2]

$$\{\hat{X}_t,\hat{M}\} \triangleq \arg\max_{\{X_t,\ M\}}(P(X_t,M,U_t,Z_t,D_t)) \tag{1}$$

Related notations are listed as

- $X_t = x_0,\ldots,x_t$: The robot's poses up to time t.
- $M = m_0,\ldots,m_k$: All known landmarks.
- $U_t = u_1,\ldots,u_t$: The control vectors up to time t.
- $Z_t = z_0,\ldots,z_t$: The observation vectors up to time t.
- $D_t = d_0,\ldots,d_t$: All decisions on association up to time t.

The SLAM problem is more complex in dynamic environments, because parts of objects are not permanently stationary: $M = M_s \cup M_d$, where M_s and M_d denote static and dynamic objects, respectively. However, the topological structure of the SLAM problem does not change in dynamic environments.

2.2. Shunting short term memory model

Arising from study of adaptive behavior, the additive Short Term Memory model is defined as follow [15]:

$$\frac{dx_i}{dt} = -A_i x_i + \sum_{j=1}^{N} f_j(x_j)B_{ji}z_{ji}^{(+)} - \sum_{j=1}^{N} g_j(x_j)C_{ji}z_{ji}^{(-)} + I_i \tag{2}$$

Eq. (2) describes the way to understand how the behavior of individuals adapts stably in real-time to complex and changing environmental contingencies. In the equation, $(-A_i x_i)$ is a passive decay; $(\sum_{j=1}^{N} f_j(x_j)B_{ji}z_{ji}^{(+)})$ is the positive feedback and $(\sum_{j=1}^{N} g_j(x_j)C_{ji}z_{ji}^{(-)})$ is the negative feedback; (I_i) is an input inspiration [15].

Adapting from Eq. (2), shunting STM equation is defined as [15]

$$\frac{dx_i}{dt} = -A_i x_i + (B_i - x_i)\left[\sum_{j=1}^{N} f_j(x_j)C_{ji}z_{ji}^{(+)} + I_i\right] - (x_i + D_i)\left[\sum_{j=1}^{N} g_j(x_j)E_{ji}z_{ji}^{(-)} + J_i\right] \tag{3}$$

where x_i is the activity of the i-th neuron; A is a decay rate; $[-D_i,B_i]$ is the boundary of the i-th neuron; $f_j(x_j)$ and $g_j(x_j)$ are the nonlinear state-dependent signal functions; C_{ji} and E_{ji} are connection strength;

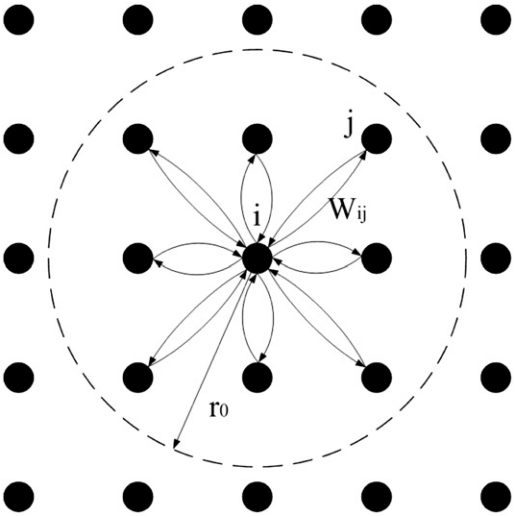


Fig. 2. Topological structure of shunting neural network. Neurons are locally related and update of neuron activities is parallel. r_0 is the impact radius; w_{ij} is the weight between the i-th and the j-th neurons.

$z_{ji}^{(+)}$ and $z_{ji}^{(+)}$ are long term memory trace; I_i and J_i are input inspirations.

Topologically, neurons in shunting networks can be locally and laterally connected, as Fig. 2 shows.

3. Shunting STM based solution to SLAM in dynamic environments

3.1. Shunting STM based map representation

There are a lot of popular map representations in the robot navigation field, including polygonal map representation (for example, visibility graph), hierarchical representation (for example, quad trees), road maps, grid maps and so on [3]. From the topological structure perspective, our proposed representation is similar to the grid map: they all uniformly rasterize the environment into small squares. However, the meanings of the representations are different. In the proposed map representation, each neuron corresponds to a small square area, and the activity of a neuron indicates the possibility of existence of a landmark inside the small square. A simple conceptual sample of the proposed map representation is shown in Fig. 3.

In order to properly design our neural network to solve the SLAM problem in dynamic environments, we need to deeply understand how to probabilistically solve the SLAM problem. The state space function of landmarks in the SLAM problem can be formulated as follows:

$$m_t = f(m_{t-1}, v_t) \tag{4a}$$

$$z_t = g(x_t, m_t) \tag{4b}$$

where $f(\cdot)$ and $g(\cdot)$ are the motion model and the observation model, respectively. Because we do not predict dynamic landmarks in the proposed method, v_t is a zero variable. Follow the tradition, we define it as $v_t \sim \mathcal{N}(0, Q_t)$, and $Q_t = \mathbf{0}$, where $\mathbf{0}$ is a zero covariance matrix.

A solution to the SLAM problem generally includes two processes, the prediction process and the update process. In the prediction process, state variables are estimated according to the motion model Eq. (4a):

$$m_{t|t-1} = f(m_{t-1|t-1}, v_t) \tag{5a}$$

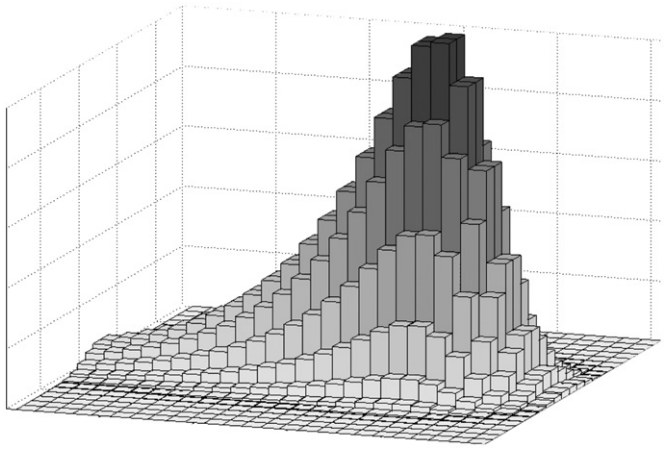


Fig. 3. A conceptual explanation of shunting STM based map representation. The figure shows the neural network; activities of neurons are indicated by height of rectangular columns. Each neuron corresponds to a rasterized grid. The activity of a neuron indicates the possibility of existence of a landmark inside that grid.

$$P_{t|t-1} = FP_{t-1|t-1}F^T + Q_t \tag{5b}$$

where P is the covariance matrix of landmarks (it equals to marginalize out all poses from the joint state variable covariance matrix); the suffix $(t|t-1)$ denotes the prediction at time point t based on the state at time point $t-1$; the suffix $(t-1|t-1)$ denotes the optimized state estimate at time point $t-1$. Because we do not explicitly predict dynamic landmarks, $f(\cdot)$ is an identity function, the Jacobian of $f(\cdot)$, F, is an identity matrix.

In the update process, state variables are updated according to the observation model Eq. (4b) and new observations:

$$m_{t|t} = m_{t|t-1} + K_t y_t \tag{6a}$$

$$P_{t|t} = (I - K_t G_t) P_{t|t-1} \tag{6b}$$

where $K_t = P_{t|t-1}G_t^T(G_t P_{t|t-1} G_t^T + R_t)^{-1}$ is the filter gain; $y_t = z_t - g(x_{t|t-1}, m_{t|t-1})$ is the innovation; G_t is the Jacobian of the observation model; R_t is the observation noise covariance matrix.

Eqs. (5) and (6) clearly explain how landmarks change in the prediction process and the update process:

$$P_{t|t-1} - P_{t-1|t-1} = Q_t \tag{7a}$$

$$P_{t|t} - P_{t|t-1} = -K_t H_t P_{t|t-1} \tag{7b}$$

Now, we design the shunting model based neural network to reflect changes described in Eq. (7).

The SLAM problem can be solved as long as our network has following characteristics.

- Activities of the neurons in the network correspond to the possibility of existence of landmarks;
- The network quantitatively describes the possibility change brought by prediction and update;
- The network becomes more and more confident on existence of static landmarks, meanwhile, it gradually "forgets" dynamic landmarks;
- Confidence on existence of landmarks grows after loop closure.

Making neurons' activities equal to probabilities can be very difficult. Fortunately, we adopt the scan matching algorithm to localize a robot in our method. Scan matching searches for the position that has high probability. Therefore, as long as the neuron with high probability always has a higher activity than the neuron with low probability, the scan matching algorithm maintains high

localization precision and high mapping consistence. This cuts loose the requirement in our method: activities of neurons are not required to be equal to probabilities, they even are not required to be proportional to them; only is monotonicity required.

It is easy to reflect the prediction process in the proposed network: just do not change the network. The update process can be simulated through passive decay and positive inspiration input.

At each update, activities of all observable landmarks update, while unobservable landmarks do not explicitly change. All observable static landmarks become more confident because of new input inspirations, while activities of all moved landmark decay and are gradually "forgot" by the network.

After a robot closes a loop, all observed landmarks receive new inspirations; therefore, confidence goes up.

The adapted shunting STM equation met the requirements mentioned above is shown in the following equation:

$$\frac{dx_i}{dt} = -Ax_i + (1-x_i)\left(I_i + \sum_{j=1}^{N} w_{ij}[x_j-\sigma]^+\right) \tag{8}$$

where parameters A denotes the passive decay rate; variable x_i is the neural activity of the i-th neuron and $x_i \in [0,1]$ is guaranteed; the excitatory inputs (inspirations) to neurons are $I_i + \sum_{j=1}^{N} w_{ij}[x_j]^+$, I_i is the excitatory input from new observations and $I_i \in [0,1]$; w_{ij} denotes weights of the excitatory connections; they are symmetric; function $[x-\sigma]^+$ is a threshold truncation function defined as $[x-\sigma]^+ = \max\{x-\sigma,0\}+\sigma$; parameter σ is the threshold of the lateral connection, and it is defined as a very small positive number. The reason we introduced σ is to improve both time efficiency and space efficiency of our method. More specifically, a too small neuron activity has no real impact on both localization and landmark representation, therefore, we abandon those very small activities to save both space and computational power.

This biological model has probabilistic meanings in the SLAM context. Parameter A denotes the speed of probability decays with time; x_i denotes the existence probability of obstacles in the i-th grid; I_i denotes the probability distribution over grids computed from the observation model; w_{ij} denotes the random probability of an object moved from the i-th grid to the j-th grid. The selection of parameter A and w_{ij} is critical. In our experiments, $w_{ij} = f(|d_{ij}|)$, where $f(a) = \mu/a$, if $0 < a < r_0$; $f(a) = 0$, otherwise; where d_{ij} represents the Euclidean distance between the i-th grid and the j-th grid. A is 0.8 in our experiments, and the sum of A and w_{ij} is less than 1. The correct selection of the parameters endows the monotonicity of the network. To confirm that the parameters are correct, we can keep adding various inspirations to a fixed position of the network, and the activities always have a bell-like shape.

3.2. Stability discussion

The stability and convergence of the proposed model Eq. (8) can be proved using a Lyapunov stability theory. We can reform Eq. (8) into the following form:

$$\frac{dx_i}{dt} = a_i(x_i)\left[b_i(x_i) - \sum_{j=1}^{N} c_{ij}d_j(x_j)\right] \tag{9}$$

through following substitutions:

$$a_i(x_i) = 1-x_i$$
$$b_i(x_i) = -A\frac{x_i}{1-x_i} + I_i$$
$$c_{ij} = -w_{ij}$$
$$d_j(x_j) = [x_j-\sigma]^+$$

Cohen and Grossberg prove that Eq. (10) is a global Lyapunov function for Eq. (9), and they also prove that if Eq. (11) holds, the network model described by Eq. (9) is Lyapunov stable [15]:

$$V = -\sum_{j=1}^{N}\int^{x_i} b_i(\varepsilon)d_i'(\varepsilon)\,d\varepsilon + \frac{1}{2}\sum_{j,k=1}^{N} c_{ij}d_j(x_j)d_k(x_k) \tag{10}$$

$$\begin{aligned}&\text{Symmetry}: \quad c_{ij} = c_{ji}\\ &\text{Positivity}: \quad a_i(x_i) \geq 0\\ &\text{Monotonicity}: \quad d_j'(x_j) \geq 0\end{aligned} \tag{11}$$

It is clear our network is symmetric by design; it is clear that our network meets the positivity requirement, for activities of neurons is bounded between 0 and 1; it is also clear that the monotonicity requirement was met, for the derivative, $d_j'(x_j)$, remains 1 in our method. Therefore, the proposed network is stable in the Lyapunov sense.

3.3. Scan matching

Traditional scan matching algorithms are applied to grid maps [3,24]. Therefore, we need to design our scan matching method to adapt to our map representation.

The goal of scan matching is to find the posterior distribution over the robot's position, $p(x_t|x_{t-1},u_t,m_t,z_t)$. The SLAM problem is modeled as a dynamic Bayesian network (see Section 2.1), therefore, we can apply Bayes' rule to reform Eq. (1) into

$$p(x_t|x_{t-1},u_t,m_t,z_t) \propto p(z_t|x_t,m_t)p(x_t|x_{t-1},u_t) \tag{12}$$

$p(z_t|x_t,m_t)$ is the observation model (Eq. (4b)); $p(x_t|x_{t-1},u_t)$ is the robot motion model. While the motion model is generally simple and is commonly formularized as a multivariate Gaussian distribution, the observation model is more complex in structure.

Our scan matching scheme estimates maximum likelihood of a robot pose based on both the pose prediction and observations. Therefore, Eq. (12) can be rewrote as

$$\begin{aligned}\log(p(x_t|x_{t-1},u_t,m_t,z_t)) &\propto \log(p(z_t|x_t,m_t)) + \log(p(x_t|x_{t-1},u_t))\\ &= \sum_{j=1}^{N}\log(p(z_{jt}|x_t,m_t)) + \log(p(x_t|x_{t-1},u_t))\end{aligned} \tag{13}$$

We have $p(z_t|x_t,m_t) = \prod_{j=1}^{N} p(z_{jt}|x_t,m_t)$ because the conditional independence was applied to the method. The conditional independence is visualized and explained in Fig. 4.

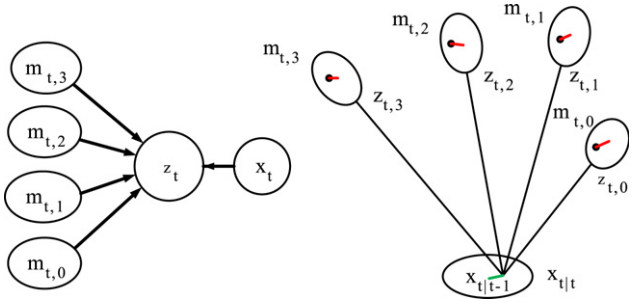


Fig. 4. Explanation to conditional independence. The left figure shows the Bayesian dependence. The right figure shows once the pose is decided, the joint likelihood shown in Eq. (13) is the sum of the pose likelihood and observation likelihoods. Both of the two kinds of likelihoods can be individually evaluated. The pose likelihood is indicated by inconsistency between the prior of pose and the posterior of pose (the green line); observation likelihoods are indicated by inconsistency between landmarks and observations (red lines). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Eq. (13) shows that the maximum likelihood estimate of a pose balances two parts: the first one is the rewards from the match between observations and corresponding landmarks; the other is the penalty from the difference between the predicted pose and the posterior estimate of the pose.

How to calculate the two parts is intuitive and simple in our method. Our map representation intrinsically probabilistically represents landmarks; furthermore, observations are probabilistic; therefore, the first part is the match between observations and landmarks, which is denoted by overlapped volumes between landmarks' distributions and observations' distributions in Fig. 5(b). The second part is simply the Mahalanobis Distance between the predicted pose and the posterior estimate of the pose.

Visibility of landmarks is desired in our method. How to achieve visibility may relate to sensor types and robot platforms. Here is how we achieve visibility information in our experiments: we compare the positions of landmarks, observations and the robot pose; if the line connected a landmark and the robot pose goes through observations, the landmark is not observable, otherwise, it is.

3.4. Complexity discussion

The computational complexity of the proposed method mainly has two sources: network update and scan matching based localization.

The adopted network is intrinsically parallel, and it has $O(m)$ complexity, where m denotes the total number of iteration times before convergence. Limited by hardware, we could not fully parallel this update. In our implementation, a Gauss Seidel based multi sequence iteration was adopted, and the corresponding complexity is $O(n)$, where n is the total number of neurons [25]. In mobile robot applications, environments are generally sparse, and we only store those non-zero sub-networks. Therefore, n is always less than one thousand in our experiments.

Scan matching is normally computationally costly, and it suffers from local maxima. GPU based parallel search and multi-resolution search can be used to decrease the computational complexity [24,3]. Besides, our method is more efficient than regular scan matching algorithms, for our map representation is naturally probabilistic, and we do not need to probabilistically rasterize the map anymore.

Our scan matching naturally adopts the sub-map technique, for at each time point, all update are performed in the observable area, which can be considered as a sub-map. Therefore, both the update process and the scan matching do not depend on the size of the map. From this perspective, the proposed method has constant computational complexity.

The space complexity of our method is lower than traditional grid maps. We only store non-zero neurons; and we do not need to duplicate the network in the update process, for we use the Gauss Seidel based iteration. At the same time, our method is not a full SLAM solution, therefore, we do not need to store all historical data.

4. Experimental results and discussion

4.1. Experiment in static environment

The proposed method is first applied to static environments.

We compare our method with the classification based EKF algorithm [3]. In that algorithm, state variables are optimized with Extended Kalman Filter [3]; data association is acquired with IPJC (Incremental Posterior Joint Compatibility Test) [9]. The classification algorithm is adopted from [8]: every time an object was re-observed, it was added as a state variable; landmarks are deleted if they cannot be observed again, and the corresponding rows and columns in the covariance matrix will also be deleted.

We apply the two methods, the classification based EKF and the proposed method, to the Victoria Park dataset, which is one of the most popular static dataset in the SLAM community [1]. The mapping results of the two methods are compared in Fig. 6. From the figure we can see that both of the two algorithms managed to converge the dataset. However, some landmarks were deleted by mistake in the classification based EKF, because there is always a lag between decision making and parameter convergence. Therefore, both localization precision and mapping consistence of the proposed method outperforms the other one (see Fig. 7 and Table 1).

Pose errors are distances between the true trajectory and the estimated one; landmark errors are distances between features' positions and landmarks' positions. Fig. 7 compares pose errors of the two methods; Table 1 compares statistics of errors of the two methods, including means, variance of errors and variance of position errors. Variance of errors, $var(e)$, is defined as $var(e)=\sum^{n}(e-mean(e))^2/(n-1)$, and variance of position errors is defined as $var(p)=\sum^{n}e^2/(n-1)$, where e are pose errors and landmark errors; n is the total number of errors.

Means and variances of position errors directly reflect localization and mapping precision; variances of errors reflect consistence. The more these parameters are close to zero, the more precise and consistent the algorithm is. Both Fig. 7 and Table 1 support that the proposed method outperforms the classification based EKF on precision and consistence in the Victoria Park dataset.

4.2. Experiment in simulated dynamic environment

Next, we apply the two methods to simulated dynamic environments.

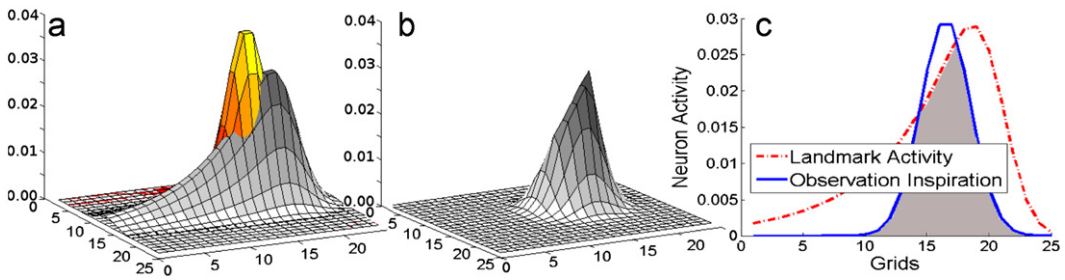


Fig. 5. Rewards from match between observation and landmark. (a) compares neurons' activities (indicated by gray) and an observation input (indicated by color); (b) shows the overlapped column; (c) shows the overlapped area in the $Y-Z$ section plane. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

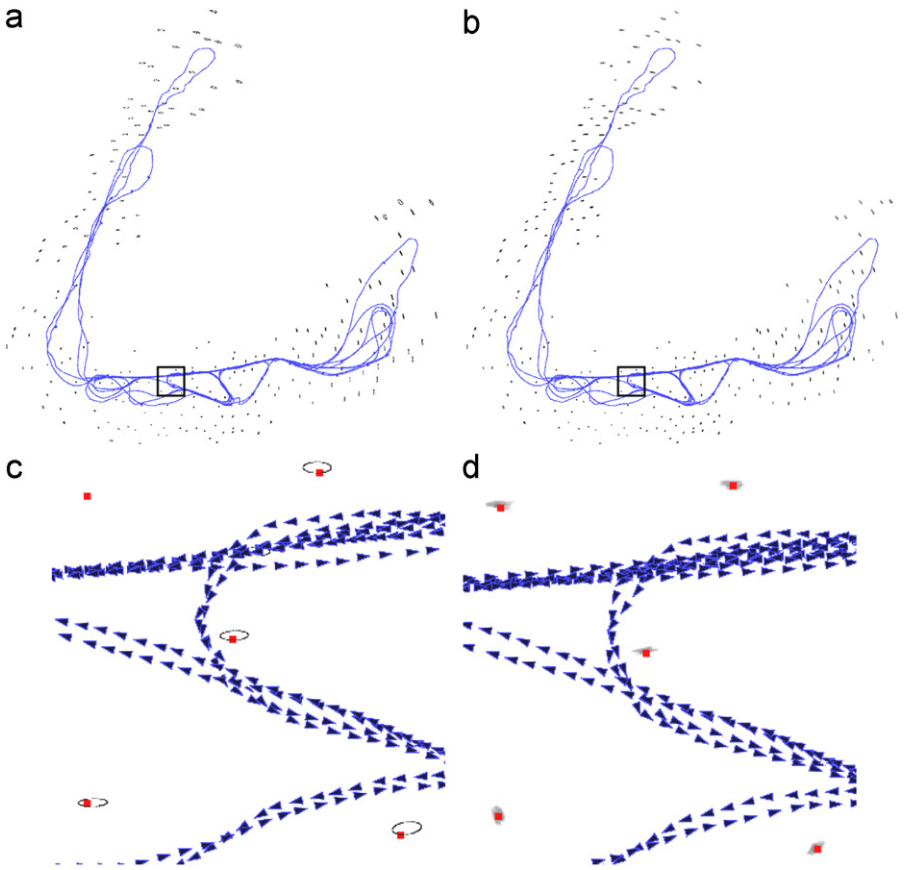


Fig. 6. Mapping results of the Victoria Park dataset. In the figure, blue triangles indicate poses of the robot; black ellipses indicate 3δ uncertainties of landmarks in the classification based EKF solution, where δ denotes the standard deviation; gray blobs indicate activities of the proposed neural network. (a) and (b) show the mapping results of the whole dataset by the two methods. The black squares in (a) and (b) are zoomed in as (c) and (d). The figure shows the classification based solution deletes some landmarks by mistake, which decreases precision and consistence, and may lead to divergence in ambiguous environments. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

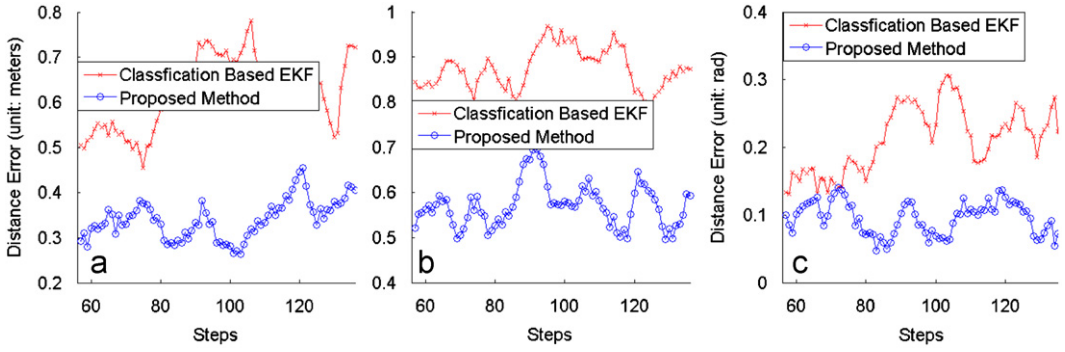


Fig. 7. Localization error comparison in the Victoria Park dataset. (a–c) show the comparison of localization errors in X, Y coordinates and heading errors, respectively. The proposed method fully exploits acquired information, while the classification based EKF always has a lag between decision making and observation acquisition. Therefore, the proposed method tends to localize the robot more precisely.

In order to clearly distinguish dynamic objects from static objects, we make static objects uniformly distribute in the space, and make dynamic objects move randomly. Following the routine, robot motions and observations are Gaussian white noise, as explained in Eq. (14), where δ_{Rr} and $\delta_{R\theta}$ denote standard deviations of distance errors and heading errors of the robot (they are 0.1 m^2 and 0.03 rad^2 in our simulation, respectively); δ_{Qr} and $\delta_{Q\theta}$ denote standard deviations of range errors and bearing errors (they are 0.001 m^2, 0.001 rad^2, respectively).

$$\Sigma_R = \begin{bmatrix} \delta_{Rr}^2 & 0 \\ 0 & \delta_{R\theta}^2 \end{bmatrix}, \quad \Sigma_Q = \begin{bmatrix} \delta_{Qr}^2 & 0 \\ 0 & \delta_{Q\theta}^2 \end{bmatrix} \tag{14}$$

The mapping results in the simulated dynamic environment are shown in Fig. 8. The figure shows that both of the two methods still finished mapping the environment, but the classification based algorithm omitted static landmarks and added dynamic objects as landmarks because of misclassification. Although in this environment, the mistakes did not lead to any disastrous result, it is possibly to diverge in complex environments or highly dynamic environments (see Section 4.3). Apparently, our method produces a more consistent map than the classification based algorithm.

We also compared precision and consistence in Fig. 9 and Table 2. Because our method intrinsically tends to make fewer mistakes than the classification based algorithm, our method has better consistence and precision in the simulated dynamic environment.

Table 1
Localization error comparison in the Victoria Park dataset. Means and variances of position errors reflect precision of an algorithm; variances of errors reflect consistency. The more these parameters are close to zero, the more consistent and precise an algorithm is. The table shows the proposed method outperforms the classification based EKF on both precision and consistence in the Victoria Park dataset.

	Robot pose						Landmark position			
	X		Y		θ		X		Y	
Statistics	EKF	Our	EKF	Our	EKF	Our	EKF	Our	EKF	Our
Mean (e)	0.6209	0.3408	0.8744	0.5715	0.2119	0.1053	0.8316	0.7112	0.9793	0.8123
Var (e)[a]	0.0806	0.0424	0.0447	0.0419	0.0048	0.0024	0.1098	0.0623	0.0697	0.0699
Var (p)[b]	0.6263	0.3435	0.8755	0.5735	0.2173	0.0985	0.8982	0.6237	0.6169	0.5091

*In the table, the unit of all means of errors in X or Y coordinates is m; the unit of mean heading errors is rad; the units of corresponding variances are m^2 and rad^2, respectively.

[a] The variance of errors, $var(e)=\sum^{n}(e-mean(e))^2/(n-1)$, where e is pose errors and landmark errors; n is the total number of errors.
[b] The variance of positional errors, $var(p)=\sum^{n}e^2/(n-1)$, where e is pose errors and landmark errors; n is the total number of errors.

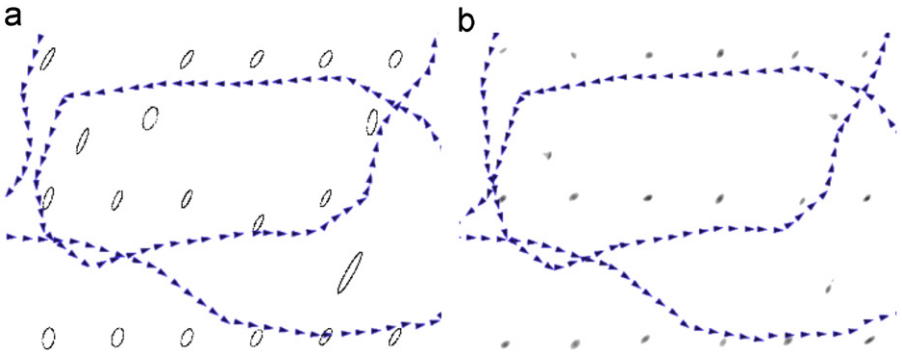


Fig. 8. Mapping results in simulated dynamic environment. (a) shows the classification based EKF mapping result, where ellipses indicate 3δ uncertainties of landmarks, where δ denotes the standard deviation; (b) shows the mapping result of the proposed method, where gray blobs indicate activities of the proposed neural network. It is clear that the proposed method produces more consistent map than the other algorithm.

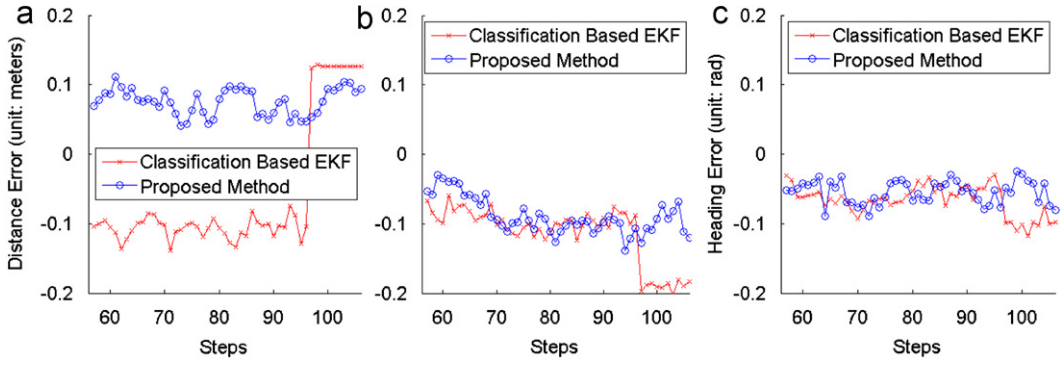


Fig. 9. Localization error comparison in simulated dynamic environment. Localization errors in X, Y coordinates and heading errors are shown in (a–c), respectively. The figure shows the proposed method has better pose precision than the classification based EKF. (a) Error in X coordinate, (b) error in Y coordinate and (c) heading error.

Table 2
Localization error comparison in simulated dynamic environment. Statistics of errors are compared in the table. Results show the proposed method outperforms the classification based EKF on both precision and consistence in the simulated dynamic environment.

	Robot pose						Landmark position			
	X		Y		θ		X		Y	
Statistics	EKF	Our	EKF	Our	EKF	Our	EKF	Our	EKF	Our
Mean(e)	−0.0598	0.0554	−0.1138	−0.0891	−0.0667	−0.0547	−0.0702	0.0692	−0.1371	−0.1029
Var(e)[a]	0.0649	0.0192	0.0412	0.0261	0.0224	0.0166	0.0899	0.0437	0.0622	0.0494
Var(p)[a]	0.1114	0.0781	0.1208	0.0927	0.0702	0.0574	0.3012	0.1191	0.2260	0.1054

[a] See the definitions and units of numbers in Table 1.

4.3. Experiment in real dynamic environment

As mentioned before, an "extremely" dynamic environment can fail the classification based EKF algorithm. We set up such an environment to verify the proposed method. In the experiment, movable chairs are the only objects in the environment, and all of them are pushed randomly in the experiment; the robot platform is equipped a laser range finder, SICK LMS-221, as the environmental sensor; visual tags [26] and ceiling cameras are used to precisely localize all objects. The environmental setup is shown in Fig. 10.

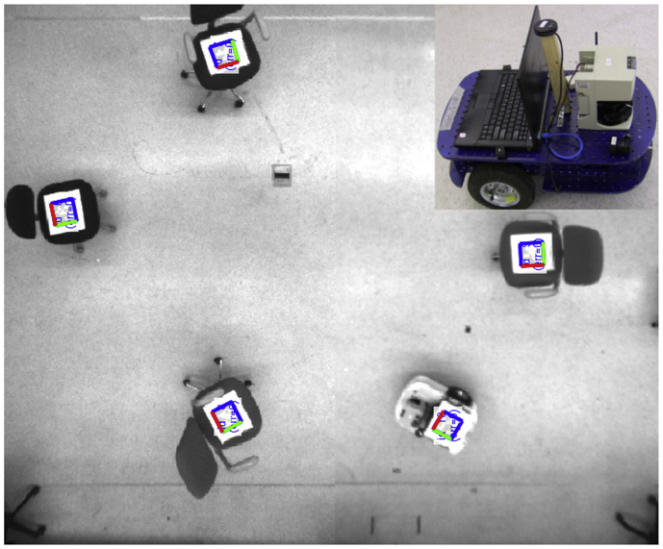


Fig. 10. Experimental platform and environment. The figure shows a sample of the experimental environment. All objects, including chairs and the robot, were labeled with visual tags, so the precise locations of objects are known all the time. All chairs are movable and are randomly moved in the experiment; a close-up of the robot is shown on the up-right corner of the figure.

In the experiment, we use a general-purpose feature extractor [27] to achieve positions and uncertainties of features, and use IPJC to perform data association.

In this highly dynamic environment, the classification based algorithm failed to converge, for it failed to acquire enough stable landmarks; our method persistently produces correct results, as shown in Fig. 11. In our method, observations are fully utilized, even inconsistence among moving objects is exploited to reject false results. Therefore, our method precisely localizes the robot and produces consistent maps in highly dynamic environments.

Again, we compared precision and consistence in Fig. 12 and Table 3. Because features in the experiment do not have stable positions, only are poses' errors counted. Because the environment is highly dynamic, almost none of parameters in the

Table 3
Pose error comparison in real dynamic environment. The table compares means and variances of errors. Statistics show the proposed has better precision and consistence than the classification based EKF in the environment.

	X		Y		θ	
Statistics	EKF	Our	EKF	Our	EKF	Our
$Mean(e)$	−0.2339	0.0762	0.4006	−0.0783	0.9674	−0.0565
$Var(e)$[a]	2.1564	0.0183	1.4014	0.0281	1.1644	0.0166
$Var(p)$[a]	2.2190	0.0781	1.4324	0.0831	1.4973	0.0592

[a] See the definitions and units of numbers in Table 1.

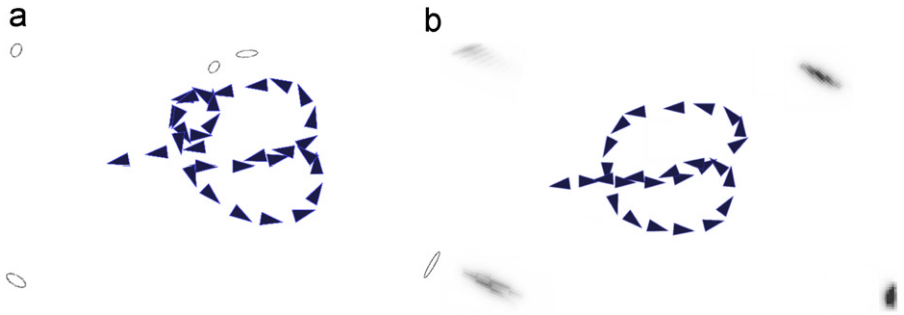


Fig. 11. Mapping results in real dynamic environment. (a) shows the classification based EKF mapping result, where ellipses indicate 3δ uncertainties of landmarks; δ denotes the standard deviation; (b) shows the mapping result of the proposed method, where gray blobs indicate activities of the proposed neural network. The classification based EKF failed to converge, while the proposed method produces consistent map and localizes precisely.

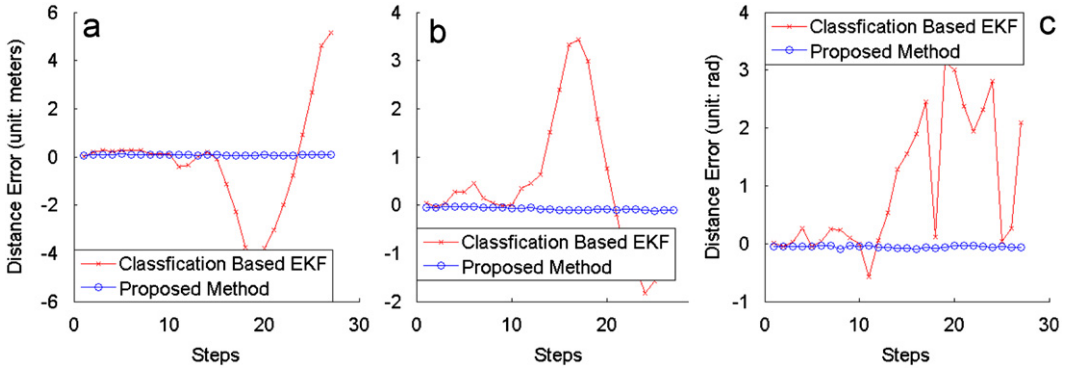


Fig. 12. Localization error comparison in real dynamic environment. (a–c) show the comparison of localization errors in X, Y coordinates and heading errors, respectively. The proposed method fully exploits acquired information, while the classification based EKF always has a lag between decision making and observation acquisition. Therefore, the proposed method succeeded on mapping and localization.

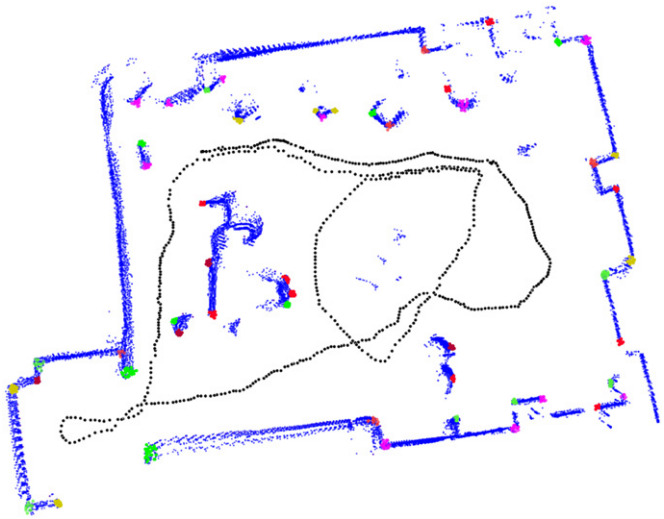


Fig. 13. Mapping result in an office. In the figure, blue points are raw LIDAR data; black points indicate the trajectory; color points are neurons with non-negligible activities (landmarks). Although there are movable objects that have very different mobility, the proposed method produces a converged and consistent map. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

classification based EKF can converge, therefore, the algorithm failed to map the environment. As a comparison, the proposed method succeeded to converge.

Lastly, we apply our method to an ordinary office environment. In the environment, chairs, people, doors and trash bins are all movable but have different mobility. The same feature extractor and data association algorithm are adopted in the experiment. The mapping result is shown in Fig. 13.

5. Conclusion

The paper presents a shunting Short Term Memory based method to solve the SLAM problem, especially in dynamic environments. The proposed method is simple: a shunting equation based map representation dynamically reflects environmental changes; a scan matching algorithm is then used to localize the robot. The proposed method is robust, efficient and has sound probabilistic foundation, because the proposed method does not discriminate objects, the scan matching process can be speed up through exploiting the map representation, and the proposed map representation can probabilistically reflects environmental changes.

Classical solutions to the SLAM problem in dynamic environments mainly focus on classification accuracy. However, objects in real environments are not always absolutely static or continuously moving. The proposed method utilizes a biological model to avoid explicitly classify objects. Each time a robot achieves new observations, the robot gets an "impression". The impression decays with time, for the longer time, the higher probability the environment changes. Each time an "impression" gets a re-match, it is inspired. Scan matching seeks the position that maximizes the match between the impression and new observations.

Theoretical analyses, software simulations and experiments in real environments, show our method is stable, robust, precise and consistent. Theoretical analyses show the proposed network has Lyapunov stability, and it has low space complexity and low computational complexity. The comparison between the proposed method and the classification based EKF in the Victoria Park dataset, simulated environments and real dynamic environments shows the proposed method outperforms the other algorithm on robustness, precision and consistence. Not only can the proposed method precisely localize the robot in various environments, it produces consistent maps as well.

In order to maintain conciseness, we do not model our method as a full SLAM solution, therefore, all historical data are abandoned. Consequently, once the method diverges, it is hard to recover from divergence. If an environment is even more ambiguous than what we showed, additional detectors for loop closure can be used to further improve the robustness of our method.

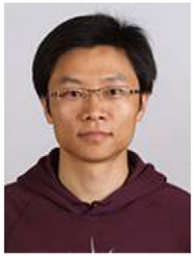
Acknowledgments

This work was supported by NSFC Grant: 61105090, 60910005.

References

[1] S. Thrun, W. Burgard, D. Fox, Probabilistic Robotics (Intelligent Robotics and Autonomous Agents), The MIT Press, 2005.
[2] H. Durrant-Whyte, T. Bailey, Simultaneous localization and mapping: part i, IEEE Robot. Automat. Mag. 13 (2) (2006) 99–110.
[3] Y. Li, Research on Robust Mapping Methods in Unstructured Environments, Ph.D. thesis, University of Science and Technology of China, Hefei, Anhui, China, May 2010.
[4] C. Bibby, I. Reid, Simultaneous localisation and mapping in dynamic environments (SLAMIDE) with reversible data association, in: Proceedings of Robotics Science and Systems, 2007.
[5] I. Miller, M. Campbell, Rao-blackwellized particle filtering for mapping dynamic environments, in: IEEE International Conference on Robotics and Automation, 2007, pp. 3862–3869. http://dx.doi.org/10.1109/ROBOT.2007.364071.
[6] J.G. Rogers III, A.J.B. Trevor, C. Nieto-Granda, H.I. Christensen, Slam with expectation maximization for moveable object tracking, in: IROS, IEEE, Taiwan, 2010, pp. 2077–C2082.
[7] D. Wolf, G. Sukhatme, Online simultaneous localization and mapping in dynamic environments, in: Proceedings of the IEEE International Conference on Robotics and Automation, 2004. ICRA'04. 2004, vol. 2, 2004, pp. 1301–1307.
[8] H. Zhao, M. Chiba, R. Shibasaki, X. Shao, J. Cui, H. Zha, SLAM in a dynamic large outdoor environment using a laser scanner, in: IEEE International Conference on Robotics and Automation. ICRA 2008, 2008, pp. 1455–1462.
[9] Y. Li, E. B. Olson, IPJC: The incremental posterior joint compatibility test for fast feature cloud matching, in: IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS 2012, IEEE, Algarve, Portugal, 2012.
[10] L. Perera, W. Wijesoma, M. Adams, Data association in dynamic environments using a sliding window of temporal measurement frames, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005), 2005, pp. 753–758.
[11] Y. Li, E. Olson, Extracting general-purpose features from lidar data, in: IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2010, pp. 1388–1393.
[12] Y. Li, E. Olson, A general purpose feature extractor for light detection and ranging data, Sensors 10 (11) (2010) 10356–10375.
[13] X. Huang, K. Hao, Y. Ding, Human fringe skeleton extraction by an improved hopfield neural network with direction features, Neurocomputing 87 (0) (2012) 99–110.
[14] C. Wang, Y. Kao, G. Yang, Exponential stability of impulsive stochastic fuzzy reaction diffusion Cohen–Grossberg neural networks with mixed delays, Neurocomputing 89 (0) (2012) 55–63.
[15] S. Grossberg, Nonlinear neural networks: principles, mechanisms, and architectures, Neural Networks 1 (1) (1988) 17–61.
[16] J.J. Hopfield, Pattern recognition computation using action potential timing for stimulus representation, Nature 376 (6535) (1995) 33–36.
[17] S. Adams, T. Wennekers, G. Bugmann, S. Denham, P. Culverhouse, Application of arachnid prey localisation theory for a robot sensorimotor controller, Neurocomputing 74 (17) (2011) 3335–3342.
[18] S. Li, S. Chen, B. Liu, Y. Li, Y. Liang, Decentralized kinematic control of a class of collaborative redundant manipulators via recurrent neural networks, Neurocomputing 91 (0) (2012) 1–10.
[19] O. Mohareri, R. Dhaouadi, A.B. Rad, Indirect adaptive tracking control of a nonholonomic mobile robot via neural networks, Neurocomputing 88 (0) (2012) 54–66.
[20] S. Li, Y. Li, Model-free control of lorenz chaos using an approximate optimal control strategy, Commun. Nonlinear Sci. Numer. Simul. (2012) 1–12.
[21] M. Milford, G. Wyeth, Persistent navigation and mapping using a biologically inspired slam system, Int. J. Robot. Res. 29 (9) (2010) 1131–1153.
[22] C. Luo, S. Yang, A bioinspired neural network for real-time concur-rent map building and complete coverage robot navigation in unknown environments, IEEE Trans. Neural Networks 19 (7) (2008) 1279–1298 http://dx.doi.org/10.1109/TNN.2008.2000394.

[23] Y. Li, M. Meng, S. Li, W. Chen, A quadtree based neural network approach to real-time path planning, in: IEEE International Conference on Robotics and Biomimetics. ROBIO 2007, IEEE, 2007, pp. 1350–1354.
[24] E.B. Olson, Real-time correlative scan matching, in: IEEE International Conference on Robotics and Automation. ICRA apos09, 2009, pp. 4387–4393. http://dx.doi.org/10.1109/ROBOT.2009.5152375.
[25] R.S. Varga, Matrix Iterative Analysis, Prentice Hall, 1962.
[26] E. Olson, AprilTag: A robust and flexible visual fiducial system, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2011.
[27] Y. Li, E. Olson, Structure tensors for general purpose LIDAR feature extraction, in: IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2011, pp. 1869–1874.



Yangming Li received B.E. and M.E. degree from Hefei University of Technology, China, in computer science, and the Ph.D. degree from University of Science and Technology of China, China, in automatic control engineering. He is currently a research associate (assistant professor) at the Robot Sensor & Human–machine Interaction Lab, Institute of Intelligent Machines, Chinese Academy of Sciences, China. His interests center on robotics, including machine perception, tracking, object classification, navigation, planning and map building.

Shuai Li received B.E. degree in precision mechanical engineering from Hefei University of Technology, China, and the M.E. degree in automatic control engineering from University of Science and Technology of China, China. Currently, he is pursing the Ph.D. degree in electrical engineering at Stevens Institute of Technology, USA. His research interests include dynamic neural networks, wireless sensor networks, robotic networks, machine learning and other dynamic problems defined on a graph. He is on the editorial board of International Journal of Distributed Sensor Networks.

Yunjian Ge received his B.E. degree from Northeastern University, China, in computer science and the Ph.D. degree from INSA, France, in computer science. He is the director of the Robot Sensor & Human–machine Interaction Lab at Institute of Intelligent Machines, Chinese Academy of Sciences, China. His interests center on robotics, including machine perception and Exoskeleton robots.