Local Map Matching Based on Fuzzy Neural Networks for Hierarchical SLAM

Xuefeng Dai, Lina Zhao, and Hui Zhang
College of Computer and Control Engineering
Qiqihar University
Qiqihar, Heilongjiang Province, 161006, China
xfdai@tom.com

***Abstract*—In order to resolve the computational complexity for local map matching of hierarchical simultaneous localization and mapping (SLAM), a novel self-organizing fuzzy neural networks (SOFNN) based approach was proposed in this paper. The matching component for local maps in the hierarchical SLAM is realized by an SOFNN. A subset of signature elements included in a local map was chosen by a clustering algorithm, then was inputted to the SOFNN. The criteria to complete a local map, and the structure learning and parameter learning algorithms for our SOFNN were discussed.**

Keywords-data association; simultaneous localization and mapping; SOFNN; neural networks

I. INTRODUCTION

The results of simultaneous localization and mapping (SLAM) developed for limited scale indoor environments were suffered from computational expensive and the huge memory space when faced to large scale environments. Improvements based on local maps were proposed, which includes the atlas framework [1], hybrid approach [2, 3], and hierarchical SLAM [4]. All of these approaches were called hierarchical SLAM broadly in this paper since they mapped the environment by a hierarchical model which the upper level is a graph to represent the topological relation of a set of local maps at the lower level. The extended *Kalman* filter (EKF) [1, 2, 4] or *Rao-Blachwellized* particle filters [3] are used for local information fusing.

Neural networks were used for mapping and localization of mobile robots. For example, in order to take away the dependence on the correct *a prior* knowledge of measurement noise covariance matrix *R*, an one input and one output neural fuzzy network was designed to adapt each diagonal element of *R* to realize adaptive *Kalman* filtering algorithm based SLAM [5]. The parameters of neural networks were trained by particle swarm optimization. It must be point out that the result was built on the fact that the plant noise model is accurate enough. On the other hand, common evidence vectors approach for voting classifications of self-organizing maps (SOM) ensemble to realize localization demonstrated improved reliability compared with individual evidence vectors for SOM ensemble [7]. Each SOM was trained on a set of robot sensor readings and afterwards the output nodes were associated with reference locations in the grid map of the environment. Nevertheless, if the environment was mapped by other modeling schemes, for example, feature based map, the approach would not take role.

The atlas framework [1] was one of the most successful approaches in dealing with SLAM in large scale environments. However, the local map matching is computational expensive. A neural fuzzy matching which can be realized by an self-organizing fuzzy neural networks (SOFNN) is preferred. Since the size and parameters of SOFNN was determined and updated by structure and parameter learning algorithms [9], respectively, so it's suitable for the matching. A revised SOFNN based on traditional ones was proposed in [6] which realized the matching according to feature parameters in each coordinates. However, it's hard for a robot to revisit each local coordinate origin exactly for the robot. A novel local map partitioning scheme and the corresponding matching approach based on SOFNN was proposed in this paper to reduce the computational complexity of matching for hierarchical SLAM.

This paper is organized as follows. Typical approaches for loop closing were reviewed in Sections 2. Section 3 discusses matching of local maps based on fuzzy rules which were realized by an SOFNN. The structure learning and parameter learning for SOFNN, and the characteristics of lower SLAM algorithm for hierarchical framework were presented in Section 4. Conclusions were given in Section 5.

II. CLOSING LOOP

When a closed loop was formed, the robot can update the global map. There're a few rules for partitioning the large scale environment into a set a subregion which will be modeled as a local map. The atlas framework divided the whole region to be mapped into a set of sub-regions. The main results of the atlas framework SLAM [1] and hybrid approach [2] for closing loop were reviewed in the following.

A three steps approach for loop closing was proposed in [1]. The core is a map matching module which takes the role is to find a match of local maps based on the elements of signature shown by Fig. 1. It's difficult to correctly closing large loop when the environment contains repetitive structure. The approach that adopted there is to defer the decision until more information is available. The number of signature

This work is supported by Science and Technology Fund of the Heilongjiang Province Education Department, China, under grant #11541389 to Xuefeng Dai.

elements to compare when matching local maps is of $O(n^2)$,

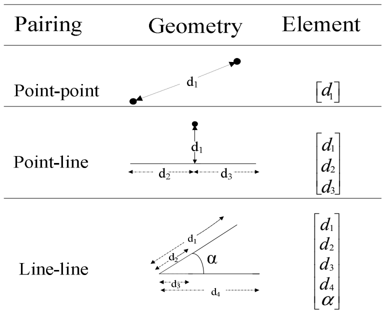

Figure 1. Local map signature elements

which may lead to $O(n^4)$.

The global map of the environment is obtained through a post-processing procedure to find a global projection of each map frame. There is a disparity v_{ij} between the transformation T_{ij} stored in the atlas graph edge and the transformation derived from the global poses of each frame:

$$v_{ij} = T_i^{\,j} \oplus T_j^0 \oplus T_0^i \ . \tag{1}$$

It was posed as a non-linear least-squares optimization problem:

$$T^* = \arg\min_T \sum_{ij} \| v_{ij} \|^2 \ . \tag{2}$$

The *Dijkstra* projection was used to compute the initial global arrangement. The readers may refer to [1] for details.

In hybrid approach, the local metric map was created by the environmental structure, for example, a room was modeled as a local map [2]. The loop closing approach differs in two aspects with other results. First, instead of closing the loops only by means of the sensor readings, loops are detected and closed by means of the topological localization. Finally, loops have to be closed only in the topological level because the local maps are a series of disconnected ones. The probability of the robot being in state *s*' after having made observation *o* while performing action *a* is give by

$$SE_{s'}(t+1) = \frac{OS(o,s',a)\Sigma_{s\in S} T(s,a,s')SE_s(t)}{P(o \mid a, SE(t))}, \tag{3}$$

where the variables were referred to [2]. *Bayesian* filtering was adopted in [3].

Local maps are sequentially built in hierarchical SLAM. Loop closing was hypothesized by an estimating of the current pose with any previous local map frame, and confirmed by RS relocation algorithm [3]. If the hypothesis can not be confirmed, loop closing was delayed until enough information was gathered.

III. Neural Fuzzy Matching

It's assumed that the upper level of the environment model of our approach is an adjacent graph representing the topological relation of the lower maps that each ones was constructed by a set of line segments. The size of a local map was determined by the number of signature elements included in each local map. The robot should know if it entered an unexplored sub-region or an explored sub-region. This was realized by the following fuzzy matching scheme.

A. Fuzzy Matching

The matching was realized by computing and comparing the signatures of each local map. It's preferred that all of the signature elements in a local map should be taken into account simultaneously. Two factors must be taken into account. One was there's such a case that a few of signature elements are similar to each other. The other was that there're total of $5C_k^2 = 2.5k!/(k\text{-}2)!$ entries included in a local map if the number of line segments included in each local map is k. In order to reduce computational burden results from the above factors, clustering algorithm was adopted here. Euclidean norm was taken as a measure of non-similarity between signature elements $\boldsymbol{s}_i$ $(=[d_{i1} \cdots d_{i4}\ \alpha_i]^T)$ and $\boldsymbol{s}_j$. If the norm

$$dist = \| \boldsymbol{s}_i - \boldsymbol{s}_j \| \tag{4}$$

is less than a threshold, then update the corresponding cluster. Else, the current input was taken as a new cluster.

The clustering algorithm was run online until the number of signature elements reached an upper bound m. Then, the current local map was completed, and a new local map was initialized at the lower level. Meanwhile, a new clustering process was initialized too.

Initially, the rulebase has no fuzzy rule. The first signature element vector

$$\boldsymbol{S}_1 = [\boldsymbol{s}_1^{\,T} \cdots \boldsymbol{s}_m^{\,T}]^T, \tag{5}$$

where $\boldsymbol{S}_1 \in R^{5m}$, $m<k$, was saved as the centre vector $\boldsymbol{C}_1$ for the first rule, i.e.,

$$c_{11}^{\ 1} = d_{11}(t), \cdots, \ c_{i4}^{\ 1} = d_{14}(t), \ c_{15}^{\ 1} = \alpha_1(t)$$
$$\cdots$$
$$c_{m1}^{\ 1} = d_{m1}(t), \cdots, c_{m5}^{\ 1} = d_{m4}(t), \ c_{m5}^{\ 1} = \alpha_m(t). \tag{6}$$

The corresponding covariance variables which resulted from line segments formed the first width vector corresponding to $\boldsymbol{C}_1$, i.e., $\sigma^1 = [\ \sigma_{11}^{\ 1} \cdots \sigma_{m5}^{\ 1}]^T$.

Afterwards, the current clusters obtained above were compared with existing clusters obtained previously by a fuzzy rulebase with the following expression

Rule i: If $d_{11}(t) = c_{11}^{\ i}$ and $\cdots$ and $d_{14}(t) = c_{14}^{\ i}$ and $\alpha_1(t) = c_{i5}^{\ i}$

and $\cdots$ and

$$d_{m1}(t)=c_{m1}^{\mathrm{i}} \text{ and } \cdots \text{ and } d_{14}(t)=c_{14}^{\mathrm{i}} \text{ and } \alpha_m(t)=c_{m5}^{\mathrm{i}},$$

$$\text{Then } o\,(t)=i \;\; (i=1,\cdots,N). \tag{7}$$

where $d_{kj}(t)$ (k=1, ⋯, m; j=1, ⋯, 5) is the ith entry of the jth signature element of the current local map, and $c_{\mathrm{kj}}^{\mathrm{i}}$ is the corresponding cluster, m is defined previously, i is the local map index and $i \in \{1, \cdots, N\}$, and N is the current size of existing local maps. The current input can be represented in a vector form

$$\boldsymbol{S}_{\mathbf{t}}=[\boldsymbol{s}_1^{\mathrm{T}} \cdots \boldsymbol{s}_{\mathrm{m}}^{\mathrm{T}}]^{\mathrm{T}}$$
$$=[s_{11} \cdots s_{15} \cdots s_{\mathrm{m1}} \cdots s_{\mathrm{5m}}]^{\mathrm{T}}, \tag{8}$$

where $\boldsymbol{s}_{\mathrm{i}}$ is the ith signature element for the case line to line given by Fig. 1, and m is the size of total signatures in a local map. *Gaussian* functions are adopted here to calculate the membership degree dgr_{i} that the current input belong to each existing local frames

$$dgr_i(S_t)=\prod_{k=1}^{m}\exp[-\sum_{j=1}^{5}\frac{(s_{kj}-c_{kj})^2}{2\sigma_{ikj}^2}] \;\; (i=1,\cdots,N), \tag{9}$$

where Π is an *and* operation which may be taken as multiply or others. If the maximal firing strength $m_{\max}=\max_{i=1,\cdots,N}\{dgr_{\mathrm{i}}(\boldsymbol{S}_{\mathrm{t}})\}$ satisfies $m_{\max}<J_{\max}$, where $J_{\max}$ is a predefined threshold, then generate a new rule, the current input and its corresponding covariance vector and the index number used to reference the current local map are taken as the fuzzy parameters for the new rule's.

Else, the current input belongs to the local map j, where

$$j=\arg\max_{i=1,\cdots,N}\{dgr_i(S_t)\}\,. \tag{10}$$

The sub-region with index j was revisited by the robot; meanwhile, parameters of the jth rule were updated. The size of rulebase is growing as the SLAM progress going on.

Only the relative angle and relative distance of two line segments are invariant to drift and rotation of robot pose. Since the signature elements were based on the above fact, so reliable and effective matching was realized.

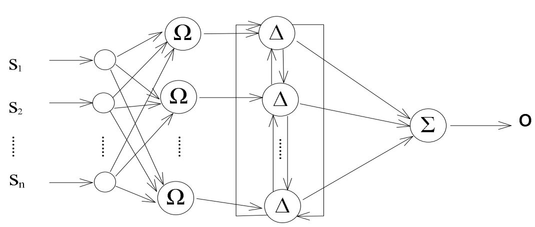

Figure 2. Architecture of PSONN

B. Architecture of SOFNN

Since the computation of (7) is time consuming, a SOFNN with four-layer feedforward architecture which was shown by Fig.2 is proposed. SOFNN have the advantages of both fuzzy reasoning and neural networks. When the number of existing local maps or the parameters of a local map varied, SOFNN can update its structure and parameters accordingly. No matter how many line segments included in each local map, the number of clustered signature elements was fixed to m, so the dimension of centre vectors and their corresponding width vectors of (8) are fixed to n=5m.

The first layer inputs the signature elements vector of the current local map. So, the input of SOFNN is a vector given by (8). The second layer calculates the membership of current input vector $\boldsymbol{S}_{\mathbf{t}}$ belongs to each existing local maps by (9). For the sake of expression simplicity, ellipsoidal basis function (EBF) neuron [9] is used here.

The third layer takes the role of lateral suppression. Each neuron in the layer was inputted an N dimensional vector. One of the input come from the ith (i=1, ⋯, N) ellipsoidal neuron, and the others come from the rest neurons in the same layer. Since the current local map can belong to at most one of the existing maps and to make the output of the most likely neuron output outstanding, the output of this layer are set to zeros except the outstanding neuron. Then, the weighted connection vector between layer 3 and layer 4 is

$$W_3^{\,4}=[1\;2\;\cdots\;N]^{\mathrm{T}}. \tag{11}$$

Finally, the fourth layer is the last layer which takes the role of selection and outputs the reasoning result. The output of the layer is a scalar variable o, which is a weighted sum of layer 3, and is given by

$$o=\sum_{j=1}^{N} w_{3\mathrm{j}}\psi_j, \tag{12}$$

where ψ_j is the output of layer 3. The architecture of SOFNN for matching is of n-N-N-1.

All of the fuzzy parameters are obtained online. The number of existing local frames determines the number of fuzzy rules, i.e. the number N of ellipsoidal neuron Ω as depicted by Fig. 2. All of the signature elements in each mapframe are stored in an ellipsoidal neuron of the SOFNN.

IV. Realization of Hierarchical SLAM

A few of issues related to the realization of hierarchical SLAM would be discussed in the following.

A. Structure learning and parameters learning

When the hierarchical SLAM algorithm runs, a series of local maps with the same number of clustered signature elements were produced. At the same time, the parameters of the global map were updated immediately after a local map was completed. In order to cover all of the existing local maps by the neural networks, the size and the parameters of SOFNN must be updated online and automatically. Structure learning determines the rulebase size, and parameter learning adjusts fuzzy parameters.

There's a slight difference between the two learning algorithms for our SOFNN and general ones. In fact, structure learning algorithm consists of two parts. One is the input space partitioning for SOFNN which was done by the signature clustering algorithm. The other is the determining of if to add an ellipsoidal neurons in the second layer, which was realized by (9). The size of layer 3 equals layer 2's. If the robot run in explored sub-region, the size of the neural network will not change, and the parameters are updated simultaneously. The size of SOFNN can be made as small as possible by deleting the neurons which correspond to deleted local maps.

Parameter learning was conducted in a partial and interval manner which was conducted as soon as the robot leaves the current sub-region. When the number of signature elements in the current local map reached the limit, all of the signature elements parameters are assigned to a new ellipsoidal neuron if the robot runs in an unexplored sub-region, and they are used to replace all of the parameters in an active neuron if the robot revisited an explored sub-region.

The set of centre vectors for SOFNN is obtained based on clustering algorithm for signature elements in each local region. Since the SOFNN was used to realize map matching, the two learning algorithms are coupled tightly with the EKF based local SLAM progress.

B. Lower level SLAM

The lower metric SLAM algorithm is an EKF based one as described in [8], [10]. It's consisted of five functional modules which are named as pose prediction, observation, measurement prediction, matching, and pose and local map update. Both of robot pose and local map are represented in a local frame. The local SLAM algorithm runs continuously and iteratively. A feature was depicted by Fig. 3 and can be represented as a vector given by

$$ls_i = [\rho_i \quad \theta_i \quad l_i \quad x_{is} \quad y_{is} \quad x_{it} \quad y_{it}]. \tag{13}$$

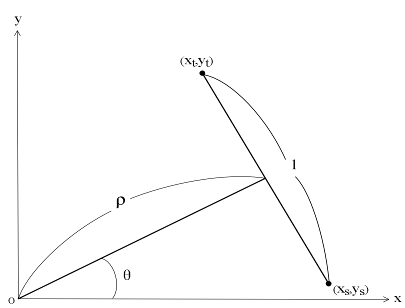

Figure 3. Demonstration of a line segment

When the lower SLAM algorithm was running, the number of line segments increased. The signature elements were calculated based on $\boldsymbol{ls}_i$ and $\boldsymbol{ls}_j$ (i, j=1, ⋯, k, $i \neq j$). The lower EKF based SLAM and the clustering algorithm are run simultaneously.

C. Other issues

If the environment has a few of similar local areas, i.e., the signature elements for two local maps are hard to differentiate, the loop closure would be confirmed by *Bayesian* localization at the upper by (3). The error correcting procedure for mapframe transition was conducted through (1) and (2) as soon as a closed-loop was formed.

V. CONCLUSIONS

A novel local map matching scheme based on SOFNN for hierarchical SLAM of large scale indoor environments was proposed. A key advantage of the scheme is the computation complexity of matching was reduced. Although it was developed for the line segment feature local map based SLAM algorithm at the lower level, it can be utilized by raw data scan and landmarks maps based algorithms as well after a moderate modification. The results should be verified by physical experiments or computer simulations. This is future work.

REFERENCES

[1] M. Bosse, P. Newman, J. Leonad, and S. Teller, "Simultaneous localization and map building in large-scale cyclic environments using the atlas framework," The International Journal of Robotics Research, vol. 23, pp. 1113-1139. December 2004.

[2] N. Tomatis, I. Nourbakhsh, R. Siegwart, "Hybrid simultaneous localization and map building: a natural integration of topological and metric," Robotics and Autonomous Systems, vol. 44, pp.3-14, July 2003

[3] J.-L. Blanco, J.-A. Fernandez-Madrigal, and J. Gonzalez, "Toward a Unified Bayesian Approach to Hybrid Metric–Topological SLAM", IEEE Transactions on Robotics, vol.24, pp.259-270, April 2008.

[4] C. Estrada, J. Neira, and J. D. Tardos, "Hierarchical SLAM: Real-Time Accurate Mapping of Large Environments", IEEE Transactions on Robotics, vol. 21, pp.588-596, August 2005.

[5] A. Chatterjee, and F. Matsuno, "A neuro-fuzzy assisted extended Kalman filter-based approach for simultaneous localization and mapping (SLAM) problems," IEEE Transactions on Fuzzy Systems, vol. 15, pp. 984-997, October 2007.

[6] Xuefeng Dai, Zhifeng Yao and Xiaomei Ning, "Fuzzy map matching for the atlas SLAM of mobile robots", The 2nd International Workshop Intelligent System and Applications (ISA2010), Wuhan, China (accepted).

[7] U. Gerecke, N. E. Sharkey, and A. J. C. Sharkey, "Common evidence vectors for self-organized ensemble localization", Neurocomputing, vol. 55, pp. 499–519, October 2003.

[8] R. Siegwart and I. R. Nourbakhsh. Introduction to Autonomous Mobile Robots (Translated into Chinese by Renhou Li). Massachusetts Institute of Technology and Xi'an Jiaotong University Press, 2005.

[9] G. Leng, G. Prasad and T. M. McGinnity, "An on-line algorithm for creating self-organizing fuzzy neural networks", Neural Networks,, vol. 17, pp.1477–1493, December 2004.

[10] Young-Ho Choi, Tae-Kyeong Lee, and Se-Young Oh, "A line feature based SLAM with low grade range sensors using geometric constraints and active exploration for mobile robot", Autonomous Robots, vol. 24, no. 1, pp. 13–27, January 2008.