Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems 2002, Lausanne

Robust Asynchronous Temporal Event Mapping

Felix Schill[*] & Uwe R. Zimmer[**]

[*]Universität Karlsruhe
Industrial Applications of Informatics and Microsystems
D-76131 Karlsruhe, Germany

[**]Australian National University
Research School for Information Science and Engineering
and the Faculty for Engineering and Information Technology
Canberra, ACT 0200, Australia

Localisation and mapping relies on the representation and recognition of features or patterns detected in sensor data. An important aspect is the temporal relationship of observations in sensor data streams. This article proposes a new approach for simultaneous localisation and mapping based on temporal relations in the flow of characteristic events in the sensor data channels.

A dynamical system is employed to acquire these correlations between simultaneous and sequential events from different sources, to map causal sequences, while considering time spans, and to recognise previously observed patterns (localisation). While this system is applicable to sensor modalities with different characteristics and timing behaviours, it is especially suitable for distributed computing. Mapping and localisation take place simultaneously in an life-long unsupervised distributed on-line learning process.

The dynamical system has been implemented as a distributed realtime system with symmetric processes. A realtime clustering network reduces the dimension of raw sensor data; cluster transitions are used as input for the dynamical mapping system. Results from physical experiments with one sensor modality are presented.

1. Motivation

Recognizing a previously seen location is extremely important for biological creatures as well as for mobile robots. The own position can be determined to some degree by dead-reckoning or odometry, but drifts are not avoidable. Re-calibration to an absolute position of the internal world model is a vital part of localisation and navigation.

Unambiguous absolute localisation from sensor input is not always possible. For only one sensor, many different locations may look the same. Results can be improved by considering other sensors, but this is not always helpful for sensors with similar characteristics. For example, if two locations look the same for a laser range finder, a sonar array may have difficulties to distinguish between them. Much more information about the absolute position can be obtained by moving around and considering the context of the current position. Assuming a place is already known, a static sensor analysis may produce multiple hypotheses about the current position. If the context or surrounding of the location is known, their number can be reduced by moving to another location. Hypotheses which are describing a different context collapse, until only the correct one is still active.

Using only local sensors (no GPS or similar techniques), it is impossible to be absolutely sure about the current position. To optimize reliability, all available sensors should be considered, as well as the context and temporal behaviour of the obtained data. This requires a method for topological mapping of events from different sensors and their temporal correlation.

A number of time series analysis methods and reproduction methods are known and well investigated. Hidden Markov models, and their more recent counterpart, the observable operator models [4] give a standard frame for temporal event mapping. Nevertheless, assuming a real-time and on-line learning context, or even worse a dynamic number of distinguishable events, neither of these methods cannot be applied directly. The computational complexity of both methods are growing (super-linearly) with the assumed number of events, and changes in the number of events result in a complete reorganization of the internal model. This makes them unsuitable for many real-time contexts. More dynamical abilities might be gained from connectionistic dynamical self-organizing approaches, for instance in the recent work from Barreto et. al. [1]. The presented temporal self-organizing network in Barreto's paper assumes a fixed number of nodes also – while the potential for a dynamical version is obvious, but which was not required in the presented example. In [3] the dynamics of a physical system have been correlated with a se-

quence of events in a planning system, but the system needs to be hand-coded.

Another common feature of all above methods is the assumed underlying rigid timing. All input data (even from different sensor modalities) are sampled and presented with a global and fixed frequency. Since each sensor data stream might and usually will have its own and flexible timing scheme in a system dealing with multi-modal real-world observations, it is reasonable to take this basic underlying asynchronism into consideration and embed it into the system. The method proposed in this article, will approach these dynamics and asynchronisms, while keeping in the frame of strict real-time constraints.

While the proposed method shares many features with the common simultaneous localisation and mapping (SLAM) approaches (see for example [2]), the fundamental difference is that there is no explicit common feature extraction (e.g. position), and any kind of involved and correlated sensor modality can be employed.

2. Network Design

The proposed asynchronous temporal event mapping system consists of two stages. The first stage of sensor clustering identifies distinguishable events in the continuous sensor data streams and delivers them to the temporal mapping network. Both stages start with an empty network, and are mainly driven by the regularities and variance in the sensor data streams.

2-1. Sensor Clustering

Let

$$\begin{aligned} Net = (C, E, R, N, \alpha) = \\ (\{c_i\}, \\ \{(c_j, c_k) \in [0, 1]\}, \\ R(c_i) \in \Re^n, \\ \alpha(c_i) \in [0, 1]), i \in 1 \ldots n \end{aligned} \tag{1}$$

be a network consisting of n cells C, edges E, representatives R (the 'centres' of the data space represented by each cell), and individual adaptation parameters α. Let α_0, λ, p, ε, h_α, h_E, t_E be the parameters controlling the network adaptation process as introduced in the following.

Assuming an empty network in the beginning, the first sensor sample S_1 is employed to generate the first cell c_1, with

$$\begin{aligned} R(c_1) &= S_1 \\ \alpha(c_1) &= \alpha_0 \approx 1/10 \end{aligned} \tag{2}$$

In each subsequent step k, the new sensor sample S_k is compared to all existing cells $\{c_i\}$, employing a metrics $\|\bullet, \bullet\|$, resulting in a distance vector D_k:

$$D_k = (d_i) = (\|R(c_i), S_k\|) \forall i \tag{3}$$

Ordering the cells in the network according to D_k, leads to a set of cells $\{c_i\}$ where

$$\|R(c_i), S_k\| \leq \|R(c_{i+1}), S_k\| \forall i \tag{4}$$

Since the process is bound to realtime constraints, the already ordered set of cells is only compared and reordered up to a certain depth, controlled by the precision parameter p, assuring for every sensor sample S_k to be at least one cell c_i with

$$\exists i; (\|R(c_i), S_k\| \leq p) \tag{5}$$

Accordingly the set of neurons is checked and reordered only up to a depth, in which a certain number m of cells can be found with

$$\exists i_1 \ldots i_m; (\|R(c_{i_j}), S_k\| \leq p \cdot \varepsilon); \varepsilon \approx 3/2 \tag{6}$$

If none close enough cell can be found in the network, i.e.

$$\neg \exists i; (\|R(c_i), S_k\| \leq p) \tag{7}$$

then a new cell is inserted:

$$\begin{aligned} R(c_{n+1}) &= S_k \\ \alpha(c_{n+1}) &= \alpha_0 \end{aligned} \tag{8}$$

and placed in the very beginning of the ordered set of cells. In the course of this insertion, the adaptation parameter α is increased for all neighbouring cells also

$$\alpha(c_i) = \alpha(c_i) + (\alpha_0 - \alpha(c_i)) \cdot \sqrt[4]{\left(e^{-\frac{i}{\lambda}}\right)}; \lambda \approx 1 \tag{9}$$

where (4) still holds. In the other case ((5) is fulfilled), the representatives are adapted according to

$$R(c_i) = R(c_i) - e^{-\frac{i}{\lambda}} \alpha(c_i) d_i \tag{10}$$

and all α_i are decreased according to their ranking

$$\alpha(c_i) = \alpha(c_i) - e^{-\frac{i}{\lambda}} \left(\alpha(c_i) - \left(e^{\frac{\log(1/2)}{h_\alpha}} \cdot \alpha(c_i) \right) \right) \tag{11}$$

with h_α determining the number of steps to halve α. After having the individual cells adapted to the most recent element of the continuous input stream, the edge-weights of the network are updated and as the result of this, cells could vanish. The edge-weights of the *previously* closest cell c_{1_p} are adapted, if the previously closest cell differs from the current one c_1:

$$(c_{1_p}, c_i) = \begin{cases} \sqrt[n_E]{e^{\frac{\log(1/2)}{h_E}}} \cdot (c_{1_p}, c_i) & \forall i \neq 1 \\ 1 & \forall i = 1 \end{cases} \tag{12}$$

with h_E determining the number of steps to halve the edge weights and n_E the number of edges emerging from the cell c_{1_p} (note that the edges are directed). All edges with a weight dropping below a threshold value t_E are considered none existent. Finally and based

on the potential recent edge elimination, cells c_i without a directed path to c_1 (criteria (13)) or without any incoming edges (criteria (14)) are deleted.

$$\neg\exists k;\forall i;(c_{k_i}, c_{k_{i+1}}) > 0$$

$$\text{where } c_{k_1} = c_i \text{ and } c_{k_n} = c_1 \tag{13}$$

$$\neg\exists j;(c_j, c_i) > 0 \tag{14}$$

This clustering system is related to the 'neural gas' network proposed in [5], but eliminates all its static restrictions. Another variation of this clustering network applied to the field of global and local spatial mapping can be found in [6].

2-2. Temporal Mapping

Given a set S_ψ of arbitrary sensors, which produce sequences of discrete events – i.e. the output of the sensor clustering stage. An event can be the occurrence of any distinguishable feature in sensor data, while a unique identifier of the currently active cluster is employed here. For mapping of causal or temporal correlations of events, a dynamical system represented by an attributed graph G_ψ is constructed for each sensor.

The nodes of G_ψ represent receptors $R_i \in R_{G_\psi}$ sensible for individual events. An occurrence of an event, which is matching the receptor results in a higher stimulation level $s_R \in [0, 1]$, while other events result in a lower stimulation level, depending on the distance of the measured input to the input expected by the receptor. Every receptor contains information about its activity level α_R and a time-stamp t_R of the last activation.

Directed internal edges $E = (R_i, R_j) \in E_{G_\psi}$ express a causal temporal relation of two events. A similar type of edges connects receptors belonging to different sensor modalities, referred to as 'cross-edges'. Their purpose is explained later in section 3.

Edges serve as 'delay lines' between receptors, with variable delay behaviour. The function $d_E(t)$ expresses the timing behaviour of events:

Definition 2.1 *Temporal function* $d_E(t)$

$$d_E: \Re^+ \to [0, 1]$$

$$d_E(t) \to e^{\left(\frac{-(t-\delta_E)}{\sigma_E}\right)^4} \tag{15}$$

d_E is symmetrically centred around δ_E; σ_E describes the tolerance or the width of the function. These two parameters are set and modified by the learning algorithm, as described in section 4-2.

Apart from the parameters δ_E and σ_E edges contain a variable learning rate parameter λ_E to control the adaptability of each edge, and a weight information $w_E \in [0, 1]$ describing the reproducibility of this transition.

The activity level α_{R_j} of a receptor R_j results from the stimulation level s_{R_j} and activity in its input edges $E_{ij} = (R_i, R_j)$:

Definition 2.2 *Receptor activity: Let* E_j *be the set of all input edges of receptor* R_j, *and* $S(E_j) \in S_\psi$ *the set of involved sensor modalities (all the sensor graphs, from which a cross edge leads to* R_j*). The receptor activity* $\alpha_{R_j}(t)$ *is calculated according to*

$$\alpha_{R_j}(t) = \phi(c_a s_R(\varepsilon + max_{E_{ij} \in E_j}\{\alpha_{E_{ij}}(t)\})) \tag{16}$$

with the edge activity

$$\alpha_{E_{ij}}(t) = w_{E_{ij}} d_{E_{ij}}(\tau_{E_{ij}}(t)(t - t_{R_i}))\alpha_{R_i}(t_{R_i}) \tag{17}$$

ϕ *is a saturating function, e.g.* $\phi(x) = 1 - e^{-x}$

The time-warping factor $\tau_{E_{ij}}$ is dynamically adapted during the recognition process. On activation of receptor R_j, the best matching input edge (R_i, R_j) with maximum activity is selected, and τ is adapted to maximize $d_{E_{ij}}(\tau(t - t_{R_j}))$. The new time-warping factor $\tau_{E_{jk}} = \tau_{E_{ij}} + c_t(\tau - \tau_{E_{ij}})$ is propagated to all output edges E_{jk} of R_j (c_t controls the speed of the time warp adaptation). This enables the network to recognize a sequence even if it is slower or faster than during the learning process.

The activation behaviour is controlled by the self-activation parameter $\varepsilon > 0$ and the amplification factor $c_a > 1$. The value for ε should be chosen close to zero. These two parameters control the generation and boost of an activity wave as a result of accurate sensor stimulation.

To classify activity, two thresholds are required. A threshold α_{high} describes the activity level which defines recognition of a sequence. A low threshold α_{low} distinguishes between inactivity and low activity. It has to be chosen higher than the activity level $\phi(c_a\varepsilon)$ resulting from self activation at stimulation.

3. Sensor Fusion

Let S be a set of sensors S_ψ which share some physical phenomena in their scan range, i.e. sensor correlation might improve the prediction qualities of individual sensor modalities, as introduced above. The idea to implement this behaviour is to extend the concept of delay edges to cross edges linking correlated modalities and representing causal and temporal relations.

Obviously the creation of these cross edges follows different rules than the creation of internal edges. First the last most active receptor R_{lma} since creation of the last cross edges is determined for each sensor modality. If its activity is higher than α_{high}, a cross edge from R_{lma} to the local most active receptor R is created. These edges are then initialized and adapted according to the rules for local edges (section 4).

Due to the additional information, which comes with the activation and later on with the construction of

prediction hypotheses, definition 2.2, needs to be extended accordingly:

Definition 3.1 *Receptor activity for multiple sensor setups: Let* $E_{S_\psi} = \{E_{ij} | R_i \in R_{S_\psi}\}$ *be the set of all input cross edges of receptor* R_j *from sensor modality* $S_\psi \in S$, *and let* $S_l \in S$; $S_l = \{S_\psi \in S | E_{S_\psi}((R_i, R_j) \neq \emptyset)\}$ *be the set of all involved sensor modalities. The local receptor activity* $\alpha_{R_j}(t)$ *is calculated according to*

$$\alpha_{R_j}(t) = \phi(c_a s_R(\varepsilon + max_{E_{ij} \in E_j}\{\alpha_{E_{ij}}(t)\} + \gamma)) \quad (18)$$

$$\gamma = c_l \tanh\left(\left(\sum_{S_\psi \in S_l}\left(\sum_{E_{ij} \in E_{S_\psi}} \alpha_{E_{ij}}(t)\right)\right) / c_M\right) \quad (19)$$

where c_l *and* c_M *control the general influence of cross edges in the system.*

c_l limits the potential influence, while c_M gives the number of modalities with maximal input for which the cross edge influence saturates.

According to (19) the additional support based on inter-modal correlations will show impact only if cross edges are actually in existence. The start-up phase in multi-model setups is therefore identical to single model systems, since activity levels need to reach a certain threshold before cross edges are created for the first time.

Multiple sensor modalities are usually handled on different processors. Thus the actual implementation of the cross edges as used for the experiments described later on, enables additional possibilities of employing different real-time network communications. While the Ada-rpc and TCP/IP showed some predictable difficulties in real-time error recovery, the finally used UDP implementation offered the highest flexibility under real-time constraints. It is therefore possible to introduce new sensor modalities at a later point in time, or to disable sensors temporary, without interrupting or even delaying the running system.

4. The Learning Process

Learning in this case has two aspects:

a. Learning new sequences by creating new receptors and edges (acquisition)

b. Modifying existing edges while recognizing known sequences (adaptation)

The adaptation process runs continuously, the acquisition process is only activated if the current input stream is not recognized (the maximum activity level in the network is below α_{high}). There is no dedicated 'learning phase'. Learning and adaptation are continuous life-long processes, controlled by the input only.

4-1. Acquisition

Whenever an event occurs in the input stream, all corresponding receptors are stimulated and evaluated. If none of them results in a activity level greater than α_{high}, a new receptor sensible for the occurred event is created. It is connected to all receptors that were highly active when the previous event occurred. The parameters δ_E and σ_E of the newly created edges are initialized according to the observed delay of events. The edge weight w_E is initialised with 1, the learning rate λ is set to a plausible start value, e.g. $1/2$. The activity level of the newly created receptor is evaluated afterwards according to definition 2.2

During this process, other hypotheses may emerge in the network. Edges are inserted from the last newly created receptor to all currently active receptors $(\alpha_R > \alpha_{low})$. When a hypothesis reaches the activity level α_{high}, the acquisition process stops.

4-2. Adaptation

Considering the realtime aspect, it is useful to adapt only those parts of the graph that are currently active. This implies that the current location of the robot is recognized. Parameters to adapt are δ_E and σ_E of the temporal function $d_E(t)$, the learning rate λ_E, and the edge weight w_E, which are all stored in the edges. Whenever a receptor R_j is stimulated by sensor data and reaches a high activity level $(\alpha_{R_j}(t) > \alpha_{high})$, those input edges of R_j are selected which have the highest edge activity level among the edges coming from the same sensor modality. The selected edges are adapted using the following rules $\Delta t = t - t_{R_i}$:

$$\delta'_E = \delta_E + \lambda_E(\tau_E \Delta t - \delta_E) \quad (20)$$

$$\sigma'_E = \sigma_E + \lambda_E(|\tau_E \Delta t - \delta_E| + t_{pmax} - \sigma_E) \quad (21)$$

$$w'_E = w_E + \lambda_E(1 - w_E) \quad (22)$$

$$\lambda'_E = c_\lambda \lambda_E \quad (23)$$

On good matches, the temporal tolerance σ_E is reduced, and the edge weight w_E increases. The constant t_{tpmax} limits the temporal precision, i.e. σ_E will not drop below this value. The reduction of the learning rate λ_E stabilises the process. The weights of the output edges of R_j are decayed by the constant c_d:

$$w'_E = w_E - \lambda_E c_d w_E \quad (24)$$

This effect is inverted in the next step for the edge which successfully activates a receptor. The weights of all other edges remain slightly reduced. Frequently occurring and successfully recognized sequences strengthen corresponding edge weights, while unused or inaccurate edges decay. This results in a probability representation coded into the edge weights, which enables the system to consider statistical information.

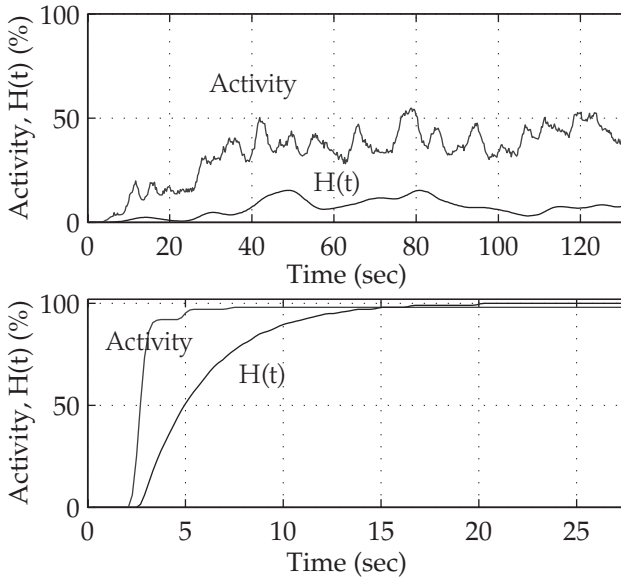

figure 1 : The upper figure shows the response of the system to a random input sequence, while the lower figure contains the response to a perfectly predictable sequence (mind the different time scales).

5. Experiments

Before testing the system in physical non-reproducible real-time experiments, two tests for basic functionality and understanding of the systems behaviours are performed. In order to evaluate the prediction qualities of the system, a function $H(t)$ is introduced, giving the ratio of successfull exact predictions of the next receptor, which will be triggered by the connected sensor in the close future. This measurement depends on the number of generated receptors in the system. As usually several hundred receptors n are generated a random pick will therefore produce a measurement $H(t)$ approximating $1/n$.

5-1. Simulated Setup

The first simulated experiment demonstrates the response to perfectly predictable and 'perfectly random' sensor sequences (figure 1). As to be expected the prediction quality is around 10% in the random case (with 10 different simulated events) and approaching 100% quickly, when a perfectly predictable sequence is presented. The activity in the random case is increasing slowly, after all possible transitions (90) were represented in the system.

The second simulated experiment demonstrates sensor correlation with a sequence of 10 events and one random bifurcation. The same sequence is presented to both modalities with different delays. The succeeding modality implements the correlation perfectly and predicts the sequence with 100% accuracy, while the preceding modality can only achieve 95% correct predictions, since it needs to guess at the bifurcation point.

The following physical experiment investigates the stability of the identification of temporal events, which are given by multiple real-world sensors and descretized through multiple dynamical clustering systems as introduced. In contrast to the simulated experiments, hundreds of events and receptors are generated here, so $H(t)$ for a random pick will be very close to zero.

5-2. Physical Setup

A simple land-bound robot was equipped with an exploration algorithm, which enables him to 'stroll' in a closed and static environment. Due to the structure of the exploration system and the usual drifts and time jitter effects, the chosen paths will never be identical, but the robot will behave similar in similar situations, which is assumed to be sufficient to produce recognizable sequences of events. Along these random paths, the raw data from laser range finders (one on-board and one mounted static in the environment), and sonar sensors is sampled and fed into the networks. Also it would of course lead to more stability to adapt the two stages of the system (clustering and temporal event mapping) in sequence, they are adapted in parallel in the following experiment, in order to test its robustness and real-world abilities.

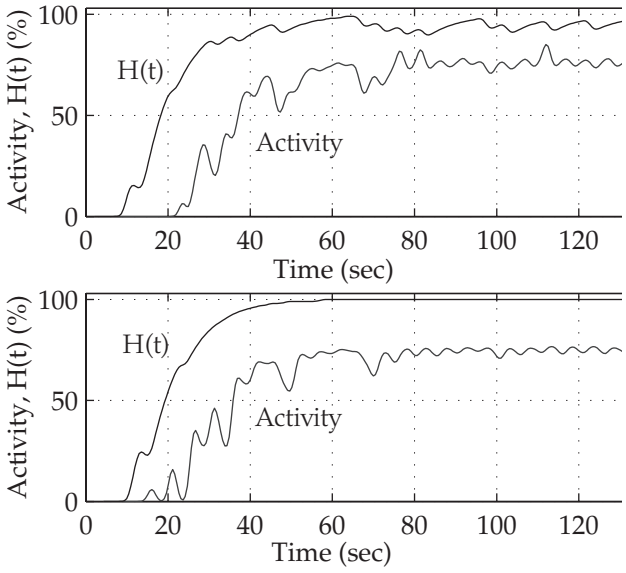

figure 2 : Activities and prediction correctness $H(t)$ for two simulated event chains, where the upper sensor modality has a random bifurcation, and the lower modality experience the same bifurcation but slightly delayed

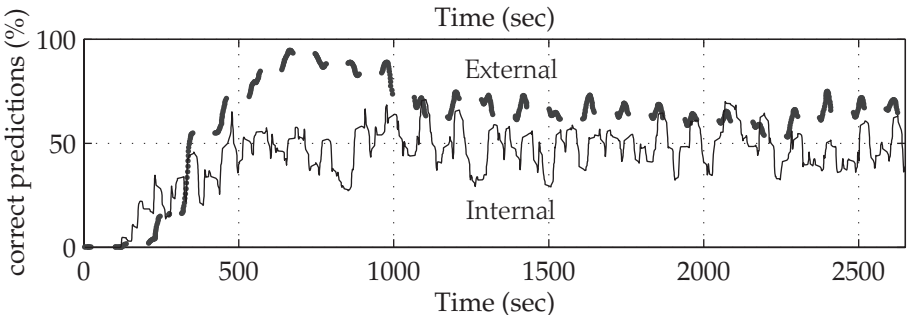

figure 3 : Physical experiment with a mobile (internal) and a fixed (external) sensor data modality. The external modality is only active when the mobile robot is in range, and is supporting the prediction quality of the internal sensor during these phases.

6. Results

The measurements shown in figure 3 indicate a number of features of the system. The prediction rate in the first three passes through the environment (up to 250 *s*) is close to zero, since the clustering systems are still unstable and the receptors in the temporal mapping network are generated (no meaningful prediction is possible while generating the underlying structure). The static external sensor (darker graph) is delivering predictions only when the mobile robot is passing through its scan range (there is nothing to predict during the static phases). While both sensors perform on average with 50-60% correct predictions (out of a set of several hundreds of potential follow up receptors), the static sensor is doing slightly better and is enhancing the results of the internal sensor during its active phases (the prediction success boosts up significantly whenever the external sensor is giving additional information via the establishes cross edges).

7. Discussion

Several hypotheses, represented by waves of activity develop and evolve in the network simultaneously. Wrong hypotheses automatically disappear, if the expected receptors are not stimulated by events any more, or if the timing is not correct. Correct ones are reinforced.

The method can recognize sequences which are disturbed by superimposed events. A wave-front is not influenced by an unexpected event, which is not represented by a receptor. As long as the next expected events from the original sequence occur in time, the activity wave continues. Sequences are also recognized, if some events are missing or falsified.

An advantage is that learning and recognition take place simultaneously. Moreover, since there is no interpretation of sensor data beyond the level of plain dynamical clustering, the introduced method does not depend on the choice of involved sensor modalities, as long as there is any correlation between them. This is usually given already by operating the sensors in the same environment. This feature separates this method from known simultaneous localisation and mapping (SLAM) methods.

The discussed real-world experiment indicates the adaptation-speed and robustness of the proposed continuously localizing and temporal event-mapping system under real-time constraints.

References

[1] Barreto, G.A.; Araújo, A.F.R.; Dücker, C.; Ritter, H.
Implementation of a distributed robotic control system based on a temporal self-organizing network
IEEE International Conference on Systems, Man, and Cybernetics (SMC'01), Tucson, Arizona, pp. 335-340., 2001

[2] Hugh Durrant-Whyte and Mike Stevens
Simultaneous localisation and mapping in complex terrains
Proceedings of the IEEE International Conference on Robotics and Automation, 2002

[3] Hertzberg, J., Jaeger, H., Morignot, P., Zimmer, U. R.
A Framework for Plan Execution in Behaviour-Based Robots
ISIC/CIRA/ISAS '98, September 14-17, 1998, Gaithersburg, Maryland, U.S.A.

[4] Jaeger, Herbert
Observable operator models for discrete stochastic time series
Neural Computation 12(6), 2000, 1371-1398

[5] Martinetz T.M., Berkovich S.G., Schulten K.J.
'Neural-Gas' Network for Vector Quantization and its Application to Time-Series Prediction,
IEEE Transactions on Neural Networks, Vol. 4, No. 4, July 1993, pp. 558-569.

[6] Zimmer, Uwe R.
Embedding local metrical map patches in a globally consistent topological map
Underwater Technology 2000, Tokyo, Japan, May 23-26, 2000