REAL-TIME ANALYSIS OF CORRELATIONS BETWEEN ON-BODY SENSOR NODES (WITH TOPOLOGICAL MAP ARCHITECTURES)

M Berchtold
TecO, University of Karlsruhe
Karlsruhe, Germany

K Van Laerhoven
University of Lancaster
Lancaster, United Kingdom

ABSTRACT

The topology of a body sensor network has, until recently, often been overlooked; either because the layout of the network is deemed to be sufficiently static (”we always know well enough where sensors are”), we always know exactly where the nodes are or because the location of the sensor is not inherently required (”as long as the node stays where it is, we do not need its location, just its data”). We argue in this paper that, especially as the sensor nodes become more numerous and densely interconnected, an analysis on the correlations between the data streams can be valuable for a variety of purposes. Two systems illustrate how a mapping of the network’s sensor data to a topology of the sensor nodes’ correlations can be applied to reveal more about the physical structure of body sensor networks.

INTRODUCTION

Large sensor networks are still a novelty, and often specifically engineered with addressable nodes that communicate their identity, purpose, and location throughout the network. Some nodes have built-in positioning systems as their applications specifically need exact locations (e.g., the Relate approach from Hazas et al., [3]), other nodes are carefully placed so that they stay arranged conform a pre-designed model. Here, we limit our investigation to a technique that estimates how close a sensor node is to another in the network by deducing this from how the sensed information from both nodes *correlates* (as previously demonstrated by Holmquist et al. [4]).

This method has not only merits for estimating spatial topologies of networks, it could also be used to search for nodes that transmit corrupted sensor data, or for tracking and dealing with changes in the topology. Related work includes the location detection of body-worn devices by Kunze et al. [7], where the location of an object on the body (such as glasses on the head, in a pocket, or in the hands) is detected by looking at the signals from built-in motion sensors during intervals where the user is walking.

COLLECTION OF DATA AND CORRELATION

The data used in this paper is taken from Van Laerhoven and Gellersen [8], where 40 accelerometers that are loosely strapped to the subject’s legs, form a distributed sensing network that is shown to be accurate enough to detect a basic range of motion-related activities such as sitting down, running, or riding a bicycle. The objective is now to get an estimation of spatial arrangement of these sensors by mere analysis of their data, and without any prior knowledge of the sensor’s location. This is not just an imaginary scenario, as this problem is closely related to that in vast wireless networks where nodes cannot be assigned unique IDs: sensor data could in that case be a complementary technique to verify where the packet of data came from.

The input data is provided by 20 sensor nodes attached to trousers (fig.1), each equipped with a 2-dimensional accelerometer sensor. It is data collected during different physical activities, such as standing, walking and climbing stairs [8]. To pre-process the data a sliding window of the size T is used. This data is

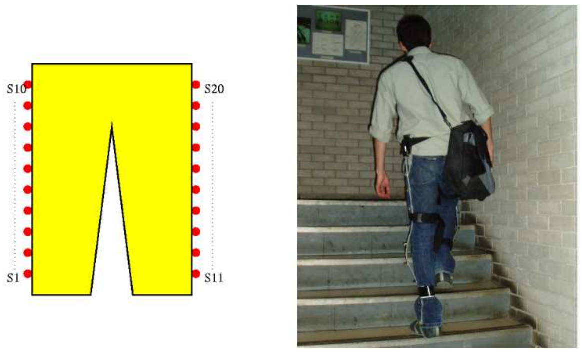

Figure 1: Distribution of sensor nodes on the trousers

stored in matrices (equation 1), one per sensor node $\mathbf{S}_k$, with two row vectors ($\mathbf{x}_l$ and $\mathbf{y}_l$) representing the dimensions of the accelerometer (1st state of figure 2).

$$\mathbf{S}_k = \begin{pmatrix} \vec{x}_k^T \\ \vec{y}_k^T \end{pmatrix} = \begin{pmatrix} x_{k0}, x_{k1}, \cdots, x_{kT} \\ y_{k0}, y_{k1}, \cdots, y_{kT} \end{pmatrix} \tag{1}$$

A correlation calculation (equation 2) is performed upon the 20 matrices of the framed input data. Each sensor node $\mathbf{S}_l$’s correlation (equation 3) towards a reference sensor node $\mathbf{S}_k$ is stored in a 40-dimensonal vector (2nd state of figure 2). The reference node selection $\mathbf{S}_k$ is shifting over the whole set of sensor nodes, which results in a cycle of 20 vectors after which the first sensor node $\mathbf{S}_1$ is the reference again. These vectors will then become input for the KSOM as shown in figure 2.

$$\begin{aligned} COR(\vec{x}_k, \vec{x}_l) &= \frac{\sum_{t=0}^{T}(x_{kt} - x_{k\mu})(x_{lt} - x_{l\mu})}{\sum_{t=0}^{T}(x_{kt} - x_{k\mu})^2 \sum_{t=0}^{T}(x_{lt} - x_{l\mu})^2} \\ &= \frac{\langle(\vec{x}_k - \vec{x}_{k\mu}), (\vec{x}_l - \vec{x}_{l\mu})\rangle}{\|(\vec{x}_k - \vec{x}_{k\mu})\|^2 \, \|(\vec{x}_l - \vec{x}_{l\mu})\|^2} \end{aligned} \tag{2}$$

$$CORR(\mathbf{S}_k, \mathbf{S}_l) = (COR(\vec{x}_k, \vec{x}_l), COR(\vec{y}_k, \vec{y}_l)) \tag{3}$$

If the nodes’ raw sensor data would be used as input for the KSOM, a lot of overwriting would occur in the map, since the different activities (standing, walking and climbing stairs) activate same regions across sensor nodes and annihilate the previous topology. To use the raw data anyway, each activity would need its own map. Another big issue is the lack of calibration for

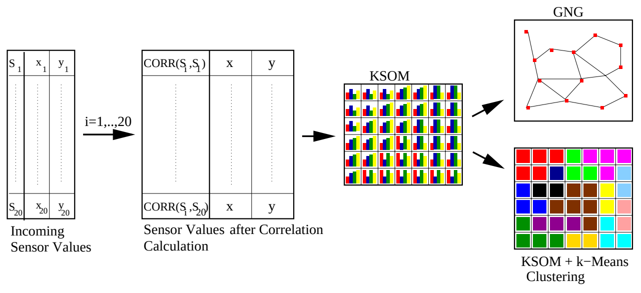
GNG
KSOM
i=1,...,20
CORR(S1,S1)
x
y
CORR(S1,S20)
Incoming Sensor Values
Sensor Values after Correlation Calculation
KSOM + k−Means Clustering

Figure 2: State flow diagram of each state of the data processing chain.

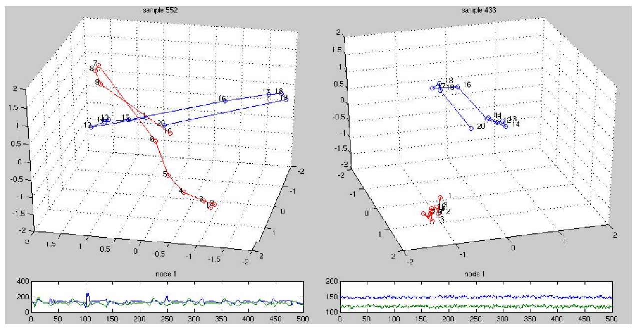

Figure 3: Independent Component Analysis (l) with much movement and (r) with nearly no movement

accelerometers. Even though some sensor nodes have nearly the same position on the trousers, they can produce a different output due to inaccuracy of the measurements. Since in the correlation calculation only similar output signals at the same time determine the outcome of the calculation, the calibration errors are eliminated. The correlation values are in the interval $[-1, 1]$, whereas $+1$ indicates total correlation, -1 if the slopes of the two waves are converse to each other and 0 if there is no correlation between the respective waves. The x and y values in the input vector of the KSOM are always alternately correlated independently (equation 3), which guarantees a better result even if the sensors are not exactly oriented towards each other. In this manner it would be best to have three dimensional sensor nodes.

INDEPENDENT COMPONENT ANALYSIS (ICA)

Our real objective can be characterised as a visualisation one: once all the incoming data signals have been correlated with each other (see the second state in figure 2), we are left with a 40 dimensional vector per sensor node that represents similarities between the signals' behaviours. Reducing this to a 2- or 3-dimensional space without losing too much of the structure in the correlation space is our core objective, to give an adequate representation of proximity between our sensor nodes: those that are strapped to the same limb segment should be in the vicinity of each other in the visualisation as well, or those that are on different legs should be further away from each other.

Obviously, we are interested in algorithms that perform this mapping from correlation space to a lower dimensional visualisation in real time, as the sensor data streams in. To have something to compare against, however, we will first show some results with a well-known off-line algorithm that obtains an ideal mapping by going over our data in multiple passes, called Independent Component Analysis (ICA). Readers interested in the inner workings of this algorithm are encouraged to use Oja's work as a starting point [5].

The correlation of the incoming sensor data delivers twenty (one per sensor node) 40-dimensional vectors that are then each plotted in three dimensional space using ICA. Figure 2 shows two typical instances of these plots, where the nodes are connected to reflect the physical arrangement in the experiment. The left plot shows the typical organisation during activities of movement (such as walking), whereas the right plot shows what happens if one of the user's legs remains immobile (and no correlation can be witnessed); The lower two plots show the real incoming acceleration data (over a history of 500 samples) for the first sensor node.

KOHONEN SELF-ORGANIZING MAP (KSOM)

The KSOM [6] is a topological map with a firm grid, in this case two dimensional. Every neuron in the map has an associated codebook vector that has the same dimension as the input vectors. As a new input vector is presented to the map, its distance to each neuron's codebook vector is calculated. The neuron with the smallest distance is the "winner" neuron. Every neuron in the grid has a relation to its neighbours, according to this relation the input data is distributed over the map, in manner of moving the codebook vectors towards the input vector correlative to the neuron's distance towards the "winner" neuron.

Since the correlation calculation produces a 40-dimensional vector, the codebook vectors need to have the same dimension. Because of visualization reasons all values have been converted to unsigned integer (math.not.:$127(x + 1)$). This high dimensionality slows down the performance of the KSOM (3rd state of figure 2) training and testing in a major way. Because of visualization, performance and mapping reasons a 25×20 map was chosen. Lower dimensionality would prevent a distinct mapping of the input vectors upon the map.

For training of the KSOM different kinds of parameter arrangements and distribution functions have been tested. The used distribution functions are a gaussian distribution and a distribution where the euclidian distance between the neurons in the grid determines the influence of the input vector on the codebook vectors. The gaussian distribution was the successful one, since the euclidian one produced only partly trained maps with totally uninfluenced regions or maps where every neuron had nearly the same codebook vector. To prevent the KSOM from over training, a linear decreasing learning rate was chosen. A set of 60000 training vectors with correlation grouped per 20 was used to train the map. In figure 6 on the left side a successfully trained KSOM is shown. The trained KSOM was used afterwards to generate a test diagram (figure 4), where the activations of neurons were counted for each sensor node as reference one.

K-MEANS CLUSTERING

The k-means clustering algorithm [1] establishes k clusters upon a set of vectors. The algorithm starts with k cluster centers, each represented via a predefined vector. For each vector of the input set, it is decided to which cluster centere it has the smallest distance. This cluster centers' vector is then moved towards the

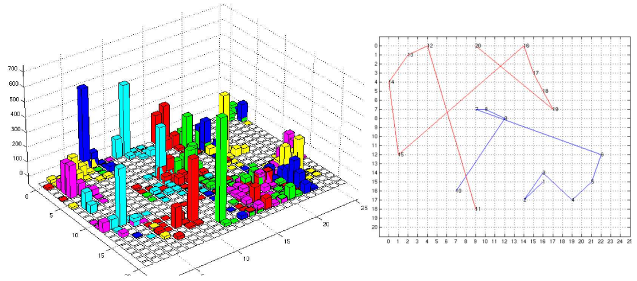

Figure 4: Test results for activations of neurons corresponding to sensor nodes. Test results for activations of neurons corresponding to sensor nodes. The left plot shows the number of times each neuron in the map has won, per sensor node (represented by differently coloured blocks). The right plot shows the top view of the same plot, connecting the winner neurons conform figures 1 and 3.

input vector by a small fraction, which is shown in equation 4.

$$\vec{x}_i^{t+1} = \vec{x}_i^t + \alpha(\vec{y}_{input} - \vec{x}_i^t), i \in \{1, ..., k\} \quad (4)$$

In our case, the 20-means clustering (one cluster per reference sensor node) is placed upon the KSOM (4th lower state of figure 2), by using the mostly over all winner neuron's codebook vector, of each sensor nodes' test set, as initial cluster center. All neuron's codebook vectors are then presented to the k-means algorithm to update the cluster centers. The cluster to which the respective neuron belongs to is visualized through different background colors, as in figure 5 can be seen.

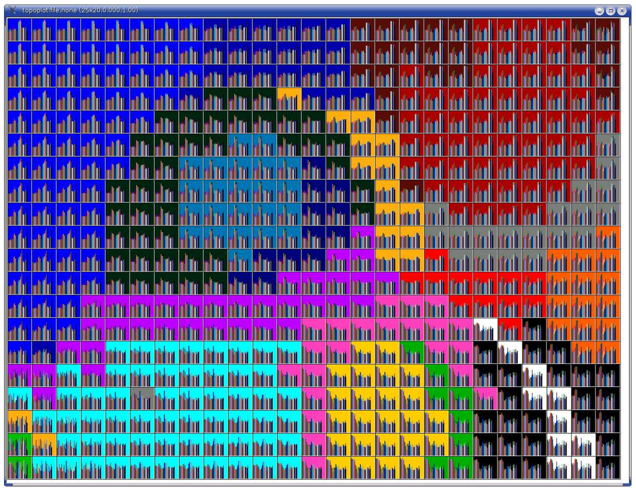

Figure 5: KSOM with k-Means Clusters

For each vector of a test set it is now decided to which cluster the corresponding winner neuron belongs to.

GROWING NEURAL GAS (GNG)

The GNG network is similar to the KSOM in manner of neurons, codebook vectors and neighbourhood relations. The differences are despite of these essential, since there is a growing number of neurons and a dynamic neighbour relationship (connections between the neurons). The GNG algorithm starts with two neurons and a connection between them. Step by step the amount of neurons is increased until the input data is satisfiably covered. GNG learns the input data similar to the KSOM via moving codebook vectors towards the input vector, but only for the two nearest neurons. For further informations on the GNG network the notes of Fritzke [2] give a precise description of its algorithm.

With a trained KSOMs' test output a two dimensional data stream is gained, since the "winner" neuron has a two dimensional position in the KSOM grid. This data stream is used to train a GNG network (4th state of figure 2) with two dimensional codebook vectors. Since the codebook vectors are two dimensional the GNG network is easily visualizable. Through training connections are made between neurons and deleted if redundant, which leads to a web (right side of fig.6) that represents the correlation of the sensor nodes. The increasing number of neurons

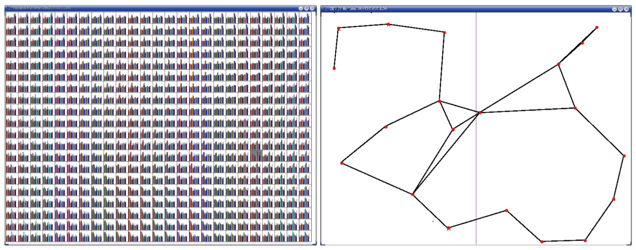

Figure 6: (l) KSOM in a final trained state (r) GNG network trained with pre-trained KSOM winner output

in the network is limited to 20, so every node in the GNG network represents one sensor node. The connections in the web form the correlations of correlated sensor nodes on the trousers.

EVALUATION

Depicted in figure 7, preliminary visual comparison of the plots from the off-line ICA-based algorithm with those from the proposed on-line algorithm shows many similarities: The proximity between nodes on lower and upper legs is present in both, and the two approaches also share the same outlier nodes (see for example S_{10}, S_{11}, and S_{20}). The plots in figure 7 are representative for most sections of the used data where sufficient motion occurred to extract the correlations among sensor nodes (note that slight motion during standing would be adequate).

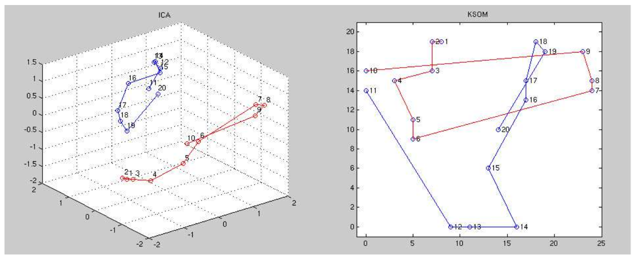

Figure 7: Topological mapping of correlations using (l) ICA in a 3D space and (r) KSOM connected winner nodes

To give a more exact measure of how the proposed topological algorithms performed on the datasets, however, we can measure how relative distances evolve over time and compare these distances with those obtained with the ICA method. Figure 8 shows the distance over time between two adjacent nodes (S_7 and S_8), while figure 9 shows the same for two nodes far away from each other in the network (S_1 and S_{20}). This particular section of the data set contained data while the test subject was standing upright, walking and climbing stairs; in the middle portion (between samples 400 and 500), the test subject stood still which resulted in poor correlation for the left leg (also observable in the right section of figure 3).

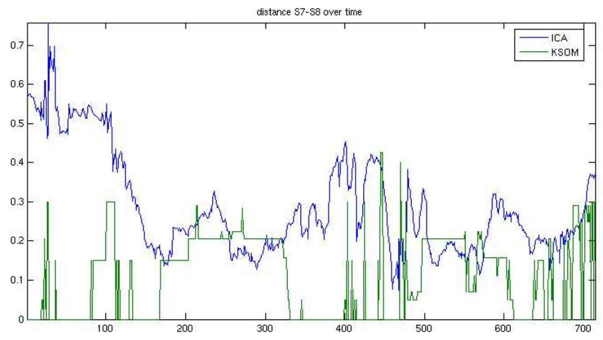

Figure 8: Distance of S_7 and S_8 over a set of 714 samples in ICA and KSOM

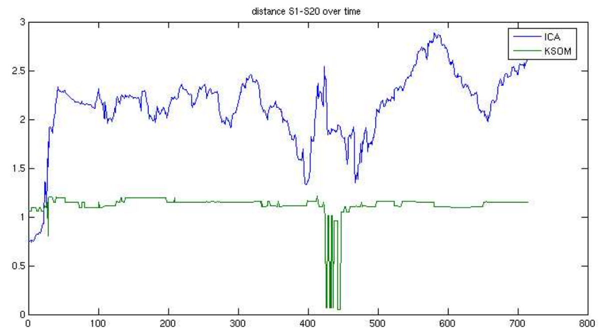

Figure 9: Distance of S_1 and S_{20} over a set of 714 samples in ICA and KSOM

Both figures illustrate that topologies indeed get stored in the Kohonen Self-Organising Map, albeit in less detail and in a more course-grained fashion than in the space obtained with ICA. It is also important to note that we used a three-dimensional ICA-created topology while our KSOM topology was restricted to a two-dimensional one, since a lean, real-time implementation was one of our goals. The real advantage of the latter is therefore preserved: it can build up a topology from the correlations as the data gets presented to the system.

Evaluating the performance of the Growing Neural Gas and K-Means clustering for additional visualisation is highly dependent on that of the KSOM, and is left as future work. It would for instance be possible to match up the graph model created by the GNG to the actual model (resembling that of figure 1) including the created edges, but we had difficulties deciding on an appropriate method to achieve this.

CONCLUSION

Correlation data often reveals the proximity of sensor nodes to each other because what they sense is more similar; This can be exploited even more so in large sensor networks where the same type of nodes have multiple sensors and are grouped in comparatively dense configurations. This paper used a network of 20 two-dimensional accelerometer nodes as a case study where each node covers a small surface area of a person's trousers.

We propose the use of correlation between distributed sensor nodes, combined with topological mapping algorithms, to approximate a spatial model of the sensor network. The algorithms discussed are self-organising in nature and operate in real-time on incoming sensor data. They can also be used as a complementary system to any sensor analysis system for networks of sensors that are observing a common phenomenon (e.g., parallel to a recognition system), and are particularly suited to body sensor networks.

The created topologies can be utilised to visually reveal defect or corrupt nodes while the sensing is in progress, or the state of the topology (or detected changes in it) could be used as an additional input in classification systems.

References

[1] C. M. Bishop. Neural networks for pattern recognition. *Oxford University Press*, 1995.

[2] B. Fritzke. A growing neural gas network learns topologies. *Advances in NIPS,7*, 1995.

[3] M. Hazas, H. Gellersen, C. Kray, H. Agbota, G. Kortuem, and A. Krohn. A relative positioning system for spatial awareness of co-located mobile devices and users. *submitted to MobiSys*, 2005.

[4] L. E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, and H.-W. Gellersen. Smart-its friends: A technique for users to easily establish connections between smart artefacts. *Ubicomp,2201/2001*, page 116, 2003.

[5] A. Hyvärinen and E. Oja. Independent component analysis: Algorithms and applications. *Neural Networks,13(4-5)*, pages 411–430, 2000.

[6] T. Kohonen. Self-organizing maps. *Springer Ser. in IS,30*, 1995.

[7] K. Kunze, P. Lukowicz, H. Junker, and G. Troester. Where am i: Recognizing on-body positions of wearable sensors. *LoCa*, 2005.

[8] K. V. Laerhoven and H.-W. Gellersen. Spine versus porcupine, a study in distributed wearable activity recognition. *ISWC,8*, pages 142–149, 2004.