Spatial processing of sensor network data

Demonstrator and feasibility study

Authors Version © IEEE, **Please quote as:** Jedermann, R.; Paul, H.; Lang, W.: Spatial processing of sensor network data. In: Proc. IEEE SENSORS 2017, Glasgow, Scotland, 2017, ISBN 978-1-5090-1012-7 pp. 73-75. See also http://ieeexplore.ieee.org/Xplore/home.jsp for updates versions

Reiner Jedermann
Institute for Microsensors, -actuators and -systems (IMSAS),
University of Bremen
Bremen, Germany
rjedermann@imsas.uni-bremen.de

Henning Paul
Department of Communications Engineering
University of Bremen
Bremen, Germany
paul@ant.uni-bremen.de

Walter Lang
Microsystems Center Bremen
University of Bremen
Bremen, Germany
wlang@imsas.uni-bremen.de

***Abstract*—In this paper we present a new approach to reduce the amount of required data transfers in wireless sensor networks by distributed in-network processing of spatial sensor data. The measurements of local clusters are described by a model consisting of a set of Virtual Sensors (VS). Quasi optimal positions of the VS in-between real sensors are calculated on sensor nodes, which act as cluster heads. Tests on a demonstrator network verified good data compression, an acceptable prediction error of the distributed algorithm, and computational feasibility on sensor nodes equipped with a floating-point ARM processor.**

Keywords—in-network processing; spatial modelling; distributed processing; wireless sensor networks; Kriging

I. INTRODUCTION

Observing the spatial distribution of a physical quantity, such as temperature, by wireless sensor networks (WSN) in an area entails two major challenges: first, to increase the robustness of the network against failure of sensor nodes or communication links, and second, to extend battery life by reducing communication. This is mostly seen as a matter of improved communication protocols. In this paper, we show that data processing by distributed algorithms can also make a significant contribution, particularly to the second challenge.

Local data processing on sensor nodes often focuses on analysing and modelling individual time-series data, e.g., Huang et al. [1]. Although several approaches are available for modelling spatial relations of the observed quantity, e.g., Murray-Bruce and Dragotti [2], evidence is slim that these approaches save communication effort compared with collection and central processing of the measurement data.

In our approach, we estimate a model to describe the spatial field with a reduced number of parameters instead of transmitting the full set of measurement points. In a previous publication [3], we presented a central implementation of this model, which reduced only the external communication of the network. The new distributed implementation also reduces the network internal communication by a factor between 3.7 and 8.

II. VIRTUAL KRIGING

Interpolation techniques often are used either to estimate values for positions that are not fitted with a sensor or to visualize measurement data from a WSN. If the measurements are overlaid with noise, the full measurement set includes redundant information. Our approach is based on the idea to retrieve approximately the same interpolation result by a smaller set of supporting points, called Virtual Sensors (VS), leading to an optimization problem to find optimal positions for a given number of VS with minimal root-mean-square prediction error (RMSE) as cost function. Kriging [4], a variant of Gaussian process regression, was selected as interpolation technique.

We applied a simple strategy to search for VS positions between the real sensors on embedded hardware. The search starts with a set of evenly distributed VS. The positions of the VS are modified by a small offset in positive and negative direction, step by step, for each VS and coordinate axis. The RMSE between the VS model and interpolation of all sensors is calculated for each point. The minimum of a parabola through the RMSE values of the three points (original, positive and negative offset) gives the new estimate for one VS coordinate.

A. Test data

The accuracy of spatial modelling was compared with synthetic test data. If the test data generation is restricted to an ideal diffusion process, as assumed by several other authors, including Dokmanic et al. [5], full reconstruction of the field is possible, in principle, depending on the amount of measurement noise. To create a more realistic scenario, we created test data by computational fluid dynamics (CFD) simulation to examine phenomena such as advection by fluid flow and the influence of walls, in addition to thermal diffusion. The test scenario consists of three heat sources in a water basin with a small flow [6]. Gaussian noise with RMS of σ_N=0.1 K (Kelvin) was added to the simulated field values between ±1°C.

B. Limits of spatial modelling

Physical processes are too complex for detailed process-based modelling. Instead, empirical models based on

This research was supported by the German Research Foundation (DFG) for the project 'In-network data analysis of spatially distributed quantities' under grants Pa2507/1 and Je722/1. For further information, visit www.intelligentcontainer.com.

observations are required. For a limited number of parameters, a certain prediction error cannot be avoided, even under optimal conditions such as noise-free measurements and high numbers (s_{Max}=500) of data points. The RMSE was 0.049 K according to simulations for this setup with 25 VS.

For a more realistic scenario, including measurement noise and a lower number of sensors, the model error primarily is masked by the noise. For example, for σ_N=0.1 K and s_{Max}=200, the prediction error of RMSE = 0.066 K is only 2% higher than that of a solution that directly interpolates all sensor data without any communication restriction.

III. EXAMPLE SCENARIO FOR DISTRIBUTED IMPLEMENTATION

Our approach for distributed processing is based on the assumption that the sensor nodes cannot directly communicate with a gateway or base station. Instead, the measurement data are collected on ClusterHead nodes and forwarded from there. The first data reduction occurs on the ClusterHeads (Fig. 1). In our example scenario, we divided the observed area into nine clusters; their local measurements were modelled by six VS each. A SuperCluster node collects all local models and reduces the size of the parameter set from 54 VS to 25 VS. Depending on the communication range of the sensor nodes, an alternate scenario with four clusters of 12 VS each is also possible (Fig. 2).

A. Demonstrator platform

SunSpot sensor nodes from Oracle [7] were used as ClusterHeads in our demonstrator platform. Their ability to execute JavaME (Micro Edition) code directly on an ARM9 processor simplifies programming of even complex signal processing algorithms in a platform-independent way, although their battery does not allow for long-term operation.

Instead of real-time measurements, the EndNodes played back data from the CFD simulation. Because of an insufficient number of hardware units, four SunSpot nodes were programmed to represent 50 EndNodes each. The code for the SuperCluster can be executed either on a SunSpot, or on a Raspberry Pi 3 board. In the latter case, the Raspberry Pi uses an additional SunSpot base station for communication.

The model parameters are transferred to a PC over a base station for data analyses and visualisation.

IV. EVALUATION OF PERFORMANCE OF DISTRIBUTED IMPLEMENTATION

In addition to the measurement of required CPU time on the demonstrator hardware, the accuracy of model prediction was evaluated by simulation on a PC platform. A total of 1000 runs were performed for different sensor positions and instances of the measurement noise.

A. Reduction of communication

For a total of s_{Max}=200 sensors, each of the nine ClusterHeads must manage 22 measurement values on average. Representing them by a model with six VS corresponds to a compression rate of 3.7. The SuperCluster performs a second compression step and thus reduces 200 sensors to 25 VS, equivalent to a compression rate of 8.

Forwarding high numbers of measurements is a major contributor to the total energy consumption. In earlier tests for monitoring temperature and humidity inside a banana container using 30 sensor nodes, an average active radio time per measurement interval of 5 seconds with a current consumption of 20 mA was required for stable forwarding of measurement data over the CC2420 radio by the 802.15.4 standard with the TinyOS operating system [8]. The length of radio listening periods was verified by simulation of CSMA (carrier sense multiple access) collision avoidance [8].

B. Required computational resources

The average processing CPU time for calculating the six local VS on the ClusterHeads was measured to 9.8 seconds per interval on the SunSpots. A fast version of the algorithm fixed the positions of four VS at the corners of the local area. Optimizing the remaining two VS required only 3.7 seconds.

The optimization of 25 VS for the SuperCluster could only be executed with some simplifications on the SunSpot nodes. The cost function for optimization, namely the RMSE between the VS model and interpolation of all measurements, is only calculated for a neighbourhood with a limited range.

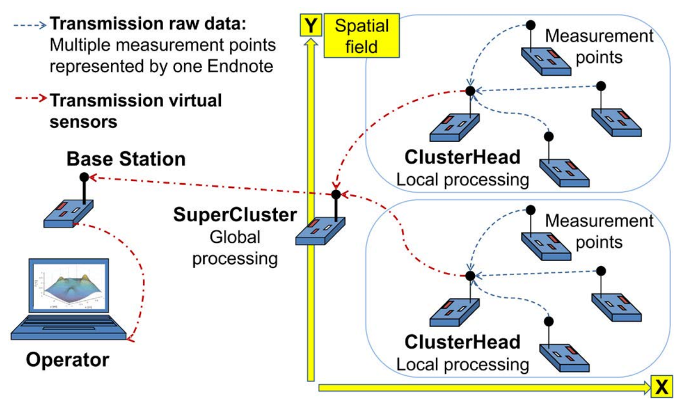

Fig. 1. Simplified set up of hardware demonstrator

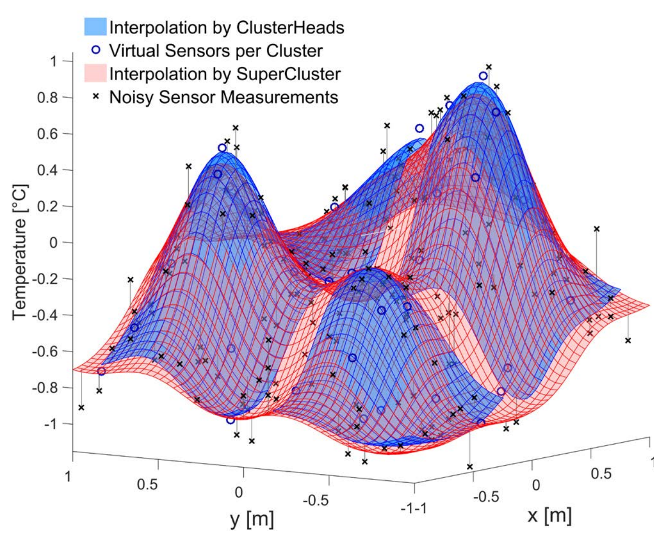

Fig. 2. Interpolation of VS for four local clusters and one SuperCluster

Even for this simplified search, the CPU time was still 130 seconds on the SunSpot, whereas the algorithm required only 0.89 seconds on the Raspberry Pi. The much faster execution can be explained by the more advanced ARMv8, including hardware floating point support, and by the optimized Java Virtual Machine (Version 1.8.0 from Oracle), which pre-compiles time-critical methods. A full search without limiting neighbour range took 4.6 seconds on the Raspberry Pi.

During execution of the algorithm, the current consumption was increased by 30 mA on the SunSpot and 130 mA on the Raspberry Pi, as measured by a shunt resistor. The required energy per measurement interval from a battery with 3 Volt was calculated and is shown in Table 1 for the variants of the algorithm. The lowest energy was required for the fast variant with the Raspberry Pi as SuperCluster with 0.35 J (Joule) and SunSpot as ClusterHead with 0.33 J. Both values are still higher than the required energy for a typical active radio time of 5 seconds with 0.3 J.

C. Prediction accuracy

The prediction accuracy of the distributed algorithm serves as the third performance criterion. The three variants of the algorithm were compared with the prediction accuracy of the central solution (RMSE 0.066 for σ_N=0.1 K and s_{Max}=200, model with 25 VS) as reference. The simplified search on the SuperCluster as standard variant increased the RMSE by 11%, the full search by only 6.7%, whereas the fast variant on the ClusterHeads led to an increase of 16%.

V. Discussion and Conclusions

The general feasibility of distributed processing of spatial sensor data was demonstrated by our test scenario. A slight increase of the prediction error must be accepted to benefit from the advantages of distributed processing such as reduction of communication.

TABLE I. PROCESSING TIME, REQUIRED ENERGY CONSUMPTION PER MEASUREMENT INTERVAL AND PREDICTION ACCURACY COMPARED TO CENTRAL SOLUTION (LAST LINE)

CPU Times and current consumption	**Variant of algorithm**		
	Full search	***Simplified search***	***Fast ClusterHead***
Rasberry Pi SuperCluster	4.6 s 1.78 J	0.89 s 0.35 J	
SunSpot SuperCluster	not feasible	130 s 11.7 J	
SunSpot ClusterHead	9.8 s 0.88 J		3.7 s 0.33 J
Increase of RMSE	6.9%	11%	16%

In our test scenario, the additional error for solutions without communication restrictions was 11% compared with a central search for 25 VS, or 13% compared with a direct interpolation of all sensor measurements.

However, the computational complexity of the related algorithms was much higher than expected. The point of break-even, when the saved energy by reduced communication is larger than the energy for computation, is still not reached with currently available sensor node hardware such as the SunSpot nodes.

A comparison of the two SuperCluster implementations shows a large variety in the computational efficiency of different ARM processor generations. Unfortunately, the Raspberry Pi is not suitable for application as a wireless sensor node due to the high stand-by current of 200 mA, primarily caused by the high number of different hardware interfaces and inability of the operating system to make use of low-power modes.

Integration of an up-to-date ARM processor into an energy-efficient sensor node would enable reduction of the energy consumption for processing of the ClusterHead algorithm to an acceptable level. Based on the availability of suitable sensor node hardware with better computational performance, distributed processing can lead to more energy efficient sensor networks.

References

[1] Y. Huang, W. Yu, C. Osewold, and A. Garcia-Ortiz, "Analysis of PKF: A Communication Cost Reduction Scheme for Wireless Sensor Networks," *IEEE Trans. Wireless Commun.*, vol. 15, pp. 843-856, 2016.

[2] J. Murray-Bruce and P. L. Dragotti, "Estimating Localized Sources of Diffusion Fields Using Spatiotemporal Sensor Measurements," IEEE Trans. Signal Process., vol. 63, pp. 3018-3031, 2015.

[3] R. Jedermann, H. Paul, and W. Lang, "Compressed radio transmission of spatial field measurements by virtual sensors," in 2016 IEEE Int. Conf. Wireless Space and Extreme Environments (WiSEE), Aachen, Germany, 2016, pp. 184-189.

[4] H. Wackernagel, *Multivariate geostatistics - an introduction with applications.* Berlin: Springer, 2003.

[5] I. Dokmanic, J. Ranieri, A. Chebira, and M. Vetterli, "Sensor networks for diffusion fields: Detection of sources in space and time," in Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on, 2011, pp. 1552-1558.

[6] R. Jedermann. (2015). *Test data for spatial field reconstruction* [Online]. Available: http://www.intelligentcontainer.com/en/downloads/spatial-test-data.html

[7] Oracle Corporation. (2016). Sun Spot World - Developers Forum [Online]. Available: http://sunspotdev.org/down.html

[8] R. Jedermann, M. Becker, C. Görg, and W. Lang, "Testing network protocols and signal attenuation in packed food transports," Int. J. of Sensor Networks (IJSNet), vol. 9, pp. 170-181, 2011.