The SIGSPATIAL Special

The SIGSPATIAL Special is the newsletter of the Association for Computing Machinery (ACM) Special Interest Group on Spatial Information (SIGSPATIAL).

ACM SIGSPATIAL addresses issues related to the acquisition, management, and processing of spatially-related information with a focus on algorithmic, geometric, and visual considerations. The scope includes, but is not limited to, geographic information systems.

**Current Elected ACM SIGSPATIAL officers are:**
- Chair, Mohamed Mokbel, University of Minnesota
- Past Chair, Walid G. Aref, Purdue University
- Vice-Chair, Shawn Newsam, University of California at Merced
- Secretary, Roger Zimmermann, National University of Singapore
- Treasurer, Egemen Tanin, University of Melbourne

**Current Appointed ACM SIGSPATIAL officers are:**
- Newsletter Editor, Chi-Yin Chow (Ted), City University of Hong Kong
- Webmaster, Ibrahim Sabek, University of Minnesota

For more details and membership information for ACM SIGSPATIAL as well as for accessing the newsletters please visit [http://www.sigspatial.org](http://www.sigspatial.org).

The SIGSPATIAL Special serves the community by publishing short contributions such as SIGSPATIAL conferences’ highlights, calls and announcements for conferences and journals that are of interest to the community, as well as short technical notes on current topics. The newsletter has three issues every year, i.e., March, July, and November. For more detailed information regarding the newsletter or suggestions please contact the editor via email at chiychow@cityu.edu.hk.

Notice to contributing authors to The SIGSPATIAL Special: By submitting your article for distribution in this publication, you hereby grant to ACM the following non-exclusive, perpetual, worldwide rights:

- to publish in print on condition of acceptance by the editor,
- to digitize and post your article in the electronic version of this publication,
- to include the article in the ACM Digital Library,
- to allow users to copy and distribute the article for noncommercial, educational or research purposes.

However, as a contributing author, you retain copyright to your article and ACM will make every effort to refer requests for commercial use directly to you.

Notice to the readers: Opinions expressed in articles and letters are those of the author(s) and do not necessarily express the opinions of the ACM, SIGSPATIAL or the newsletter.

The SIGSPATIAL Special (ISSN 1946-7729) Volume 7, Number 2, July 2015.
Table of Contents

Message from the Editor .............................................................. 1
Chi-Yin Chow

Section 1: Special Issue on Geosensor Networks

Introduction to this Special Issue: Geosensor Networks ................ 2
Silvia Nittel

Deployment of a Large-Scale Soil Monitoring Geosensor Network .................................................. 3
Gopal K Mulukutl, Brian T. Godbois, and Serita Frey

Challenges to Using Decentralized Spatial Algorithms in the Field: The RISERnet Geosensor Network Case Study .................................................. 14
Matt Duckham, Xu Zhong, and Kevin Toohey

Real-time Sensor Data Streams .................................................. 22
Silvia Nittel

Fusing Human and Technical Sensor Data: Concepts and Challenges .................................................. 29
Bernd Resch and Thomas Blaschke

Formal Foundations of Sensor Network Applications .................................................. 36
Jacob Beal and Mirko Viroli

Section 2: Event Reports

ACM SIGSPATIAL MapInteract 2014 Workshop Report ......................... 43
Falko Schmid, Chris Kray, and Holger Fritze

ACM SIGSPATIAL BigSpatial 2014 Workshop Report ......................... 44
Varun Chandola and Ranga Raju Vatsavai
Message from the Editor

Chi-Yin Chow
Department of Computer Science, City University of Hong Kong, Hong Kong
Email: chiychow@cityu.edu.hk

In the first section, we have a special issue of some topic of interest to the SIGSPATIAL community. The topic of this issue is “Geosensor Networks” which is edited by our associate editor Prof. Silvia Nittel. Prof. Nittel is currently an Associate Professor in the School of Computing and Information Science and a faculty member with the National Center of Geographic Information and Analysis at the University of Maine, Orono, USA.

The second section consists of two event reports from:

1. The 2nd ACM SIGSPATIAL International Workshop on Interacting with Maps (ACM SIGSPATIAL Map-Interact 2014)
2. The 3rd ACM SIGSPATIAL International Workshop on Analysis for Big Spatial Data (ACM SIGSPATIAL BigSpatial 2014)

I would like to sincerely thank all the newsletter authors, Prof. Nittel, and event organizers for their generous contributions of time and effort that made this issue possible. I hope that you will find the newsletters interesting and informative and that you will enjoy this issue.

You can download all Special issues from:

http://www.sigspatial.org/sigspatial-special
The SIGSPATIAL Special

Section 1: Geosensor Networks
Introduction to this Special Issue: Geosensor Networks

Silvia Nittel
Spatial Informatics, School of Computing and Information Science, University of Maine, USA

The last two decades have seen unprecedented advances in the development of small-scale sensors, as well as inexpensive, small computing platforms and a plethora of wireless communication media. These technological developments have lead to the research area of geosensor networks (GSN), which are wireless sensor networks deployed in geographic space. From a practical perspective, we see GSN often as ‘networked geosensors’ that live-stream information to the Internet today. Sensors can range from stationary environmental sensors to drones, ocean drifters, autonomous unmanned vehicles collecting environmental data or even humans acting as sensors. A GSN enables us to observe, reason about and react to events in geographic space in near real-time.

This special issue consists of five contributions that are differently related to GSN. The first two contributions explore hands-on experiences with true GSN field deployments: Gopal Mulukutla, Brian T. Godbois and Serita Frey from the University of New Hampshire describe their development of a distributed GSN to monitor soil moisture, soil CO₂ efflux and other parameters to increase scientific understanding of the complex interactions of ecological, biogeochemical and meteorological processes. The second article comes from a GSN research perspective and looks into taking decentralized algorithms from the lab to the field. Matt Duckham, Xu Zhong and Kevin Toohey discuss their experience designing and deploying a 70-node wireless GSN for monitoring environmental conditions relevant to wildfire hazard in Victoria, Australia. The authors review decentralized algorithms for understanding observations under the constraints of a realistic setting with today’s off-the-shelf hardware. Similarly, my own contribution focuses on a consequence of GSN deployments, which are massive amounts of real-time sensor data streams. These sensor data streams contribute to the problem of ‘big spatial data’ and also raise the interesting question of efficient and effective near real-time data analysis. The article discusses specific data management requirements due to massive real-time sensor data streams and provides some guidance for identifying effective data management technology, ranging from Hadoop-GIS, NoSQL tools, spatio-temporal database systems and data stream engines.

The fourth contribution by Bernd Resch and Thomas Blaschke focuses on the ever more important aspect of humans as ‘sensors’ providing relevant data without the need for expensive infrastructure, as for example in traffic monitoring with traffic jam alerts or citizen science efforts. However, information gathered by humans poses challenges such as interpreting and fusing heterogenous, unstructured and semantically rich information. In the fifth article Jake Beal and Mirko Viroli explore the much necessary aspect of formal foundations for geosensor network programming based on the field calculus.

I hope that the readers will enjoy reading this issue and appreciate the multiplicity of views it offers.
Deployment of a Large-Scale Soil Monitoring Geosensor Network

Gopal K Mulukutla\textsuperscript{1}, Brian T. Godbois\textsuperscript{2a} and Serita Frey\textsuperscript{2b}

\textsuperscript{1}Earth Systems Research Center, University of New Hampshire, Durham, NH, US. gopal.mulukutla@unh.edu

\textsuperscript{2}Department of Natural Resources and the Environment, University of New Hampshire, Durham, NH, US. 
\textsuperscript{a} briangodbois@gmail.com, \textsuperscript{b} serita.frey@unh.edu

Abstract

\textbf{We provide an overview of our practical experience with developing a distributed sensor network to monitor soil response to climate change and increase our understanding of the complex interactions of the surrounding ecological, biogeochemical and meteorological processes. The network consists of seven sites with unique topographical, and land-use characteristics, spread across a large area in the state of New Hampshire (US). The system was designed to measure soil moisture, soil CO\textsubscript{2} efflux and make other ancillary measurements (air temperature, precipitation, wind speed etc.). The system design encompasses sensor and hardware selection, customization and the overcoming design constraints such as the need to operate a power hungry sensing system at remote locations with access only to solar power. The data we collect streams to the web as an outreach and teaching resource, provides input to ecosystem models used to predict how ecosystems in the region will respond to climate and land-use change, and directly monitors soil properties and processes in a changing climate.}

1 Introduction

Field measurements are a critical component to our increased understanding of the environment. Studying soils and processes occurring in them allow us to increase our understanding of ecological, biogeochemical and meteorological processes occurring in the surroundings. Carbon stock present in soil is a critical component of the global carbon (C) cycle and thus affects global climate, while soil moisture plays an important role in energy and water cycles by regulating the interaction of the land surface with the atmosphere ([2],[1]). The challenge of studying and quantifying many of these processes is complicated by the role spatial scale plays in driving variability, and the complex interactions among factors such as climate and land-use across spatial scales. For example, soil moisture at small spatial scales is dictated by factors such as topography, soil type, vegetation type, root structure etc., while at larger spatial scales atmospheric conditions play an important role. Moreover soil moisture variation with depth is an important parameter for understanding ecosystem water balance that only field measurements can capture and techniques such as satellite remote sensing cannot provide a complete
picture. Current satellite measurements are estimated to detect soil moisture only in the 1-2 cm depth range; whereas various processes can affect variability of soil moisture across the soil column (for example vegetation can extract moisture from deeper soil layers) ([1]).

As part of a state-wide initiative in New Hampshire to monitor ecosystems in response to climate and land-use change, an innovative, integrated statewide system of sensors was built to support research aimed at understanding the complex interactions among climate, land-use and society. The project focuses on the many services provided by ecosystems in the state: recreation and tourism, carbon sequestration, regional climate regulation, biomass for electricity generation, and pollutant removals from air, soil and water. Data from these sensors is being used to parameterize and validate a suite of climate, hydrological and ecosystem models over the extended statewide domain to predict changes in ecosystem function and understand their effects for society ([4]). The primary objective of the soil sensor network is to monitor soil temperature, moisture, respiration ($CO_2$ flux), and related environmental and meteorological parameters across representative land-uses. The sensing systems were deployed at seven locations across the state (summarized in Table 1). This paper describes system design, sensor and hardware selection and customization, and the challenge of developing and optimizing a power-hungry sensing system running on solar power at four of the locations.

### Table 1: Summary of seven sites chosen for the deployment of soil monitoring network.

<table>
<thead>
<tr>
<th>No.</th>
<th>Site Name</th>
<th>Land-use</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Hubbard Brook Experimental Forest (HBF)</td>
<td>Higher Elevation Pristine Northern Hardwood Forest</td>
<td>Thornton, NH</td>
</tr>
<tr>
<td>2.</td>
<td>Burley-Demeritt Farm (BDF)</td>
<td>Pasture</td>
<td>Durham, NH</td>
</tr>
<tr>
<td>3.</td>
<td>Thompson Farm (THF)</td>
<td>Uneven Aged Mixed Forest and Pasture</td>
<td>Durham, NH</td>
</tr>
<tr>
<td>4.</td>
<td>Dowst-Cate Town Forest (DCF)</td>
<td>Mixed forest with ongoing logging operations</td>
<td>Deerfield, NH</td>
</tr>
<tr>
<td>5.</td>
<td>Bartlett Experimental Forest (BRT)</td>
<td>Higher Elevation Northern Hardwood Forest With a history of logging</td>
<td>Bartlett, NH</td>
</tr>
<tr>
<td>6.</td>
<td>Saddleback Mountain (SDM)</td>
<td>Higher Elevation Mixed Forest with a history of clear cutting</td>
<td>Deerfield, NH</td>
</tr>
<tr>
<td>8.</td>
<td>Blue Hills near Trout Pond Brook (TPB)</td>
<td>Mixed Transitional Forest</td>
<td>Strafford, NH</td>
</tr>
</tbody>
</table>

2 Soil Monitoring

2.1 Soil Respiration

Soil respiration (SR) is the efflux of carbon dioxide from soils that is a result of a complex suite of below ground biological and physical processes involving plants, microorganisms, and soil constituents (e.g. minerals) ([7]). $CO_2$ efflux from soils is a major component of the global carbon budget and enhanced $CO_2$ fluxes due to environmental change may provide feedbacks to the climate system. SR is a complex process, highly sensitive
to temperature, moisture, and human intervention (e.g. land use change) among others ([6]). Continuous field measurements of \( CO_2 \) efflux are critical to understanding the underlying processes involved and how ecosystems will respond to climate change.

SR is highly variable at different time scales: sub-hourly, daily to seasonal, annual, and inter-annual. A widely used way to measure the areal rate of SR is by determining soil \( CO_2 \) efflux from dynamic chamber measurements over an area. \( CO_2 \) flux at a point is measured by placing a chamber over soil. Variation in \( CO_2 \) concentration within the headspace of the chamber is recorded and used to estimate a flux rate. Similar area-averaged measurements using multiple chambers spread over an area are used to determine areal flux and SR rates. A statistically significant number of chambers are required to gather data representative of a site, along with a robust, high temporal frequency sensing system capable of long term measurements with minimal downtime.

2.2 Soil Moisture

Soil moisture (SM, also referred to as volumetric water content) is the amount of water contained in soil. Almost all hydrological, biological or biogeochemical processes occurring in the soil column are affected by this parameter, making it an important indicator of ecosystem health ([5]). SM is usually determined by measuring a surrogate, dielectric permittivity, using principles of time-domain or frequency-domain reflectometry (TDR or FDR) ([3]). An FDR sensor consists of multiple prongs parallel to each other that are inserted into the soil. An oscillating electromagnetic wave sent to the prongs charges according to the dielectric permittivity of the soil. The stored charge that is proportional SM is measured. A TDR sensor consists of two metal prongs inserted at a certain depth in soil. The velocity of an electromagnetic wave propagated along the probe rods depends on the dielectric permittivity of soil surrounding the prongs. Increasing SM reduces the propagation velocity due to an increase in dielectric permittivity. The two-way travel time of the signal is measured and related to SM.

2.3 Ancillary Measurements

Weather conditions and other related parameters such as soil temperature play an important role in how SM and SR vary over time and space by affecting chemical reactions, nutrient turnover and microbial metabolism. We implemented the sensing system design to include the following variables: soil temperature, air temperature, precipitation, snow depth, wind speed and direction.

3 Sensor Node Requirements

3.1 Site Selection

Site selection required careful consideration of factors such as topography, vegetation, soil-type etc. Another important consideration was the ability to co-locate a water quality monitoring network. This was facilitate interdisciplinary research aimed at understanding the hydrological and biogeochemical linkages between the terrestrial and aquatic environments. Table 1 summarizes the location of each site with specific land-use characteristics highlighted, and Figure 1 highlights the location of these sites on a map. Six of the seven sites are within forested headwater catchments and in close proximity to a water quality monitoring network.
4 System Architecture

We provide a description and functionality of individual component that make up a sensing system node. A sensor system node consists of all associated sensing equipment at one specific location within the site.

1. The automated chamber and its related hardware.
2. The multi-channel gas sampling system
3. Compression system and automated chamber control
4. The soil moisture sensing system
5. Ancillary measurements system.
6. Datalogger, system control and telemetry

4.1 Multi-channel gas sampling

Each site consists of 6 to 8 dynamic automated chambers installed as part of a single sensing system. Each custom-built chamber unit (Fin-Landis Techniker LLC, Nottingham, NH), consists of a stainless steel frame with a collar, and an aluminum chamber lid (46 cm length and width, and 20 cm in depth) (Figure 2). The frame is inserted into the ground with the collar resting evenly on the surface. The chamber lid is hinged to the collar and is controlled by a pneumatic actuator arm which opens and shuts based on the flow of compressed air controlled by a compression system. Chamber closing creates incubation within the enclosure to enable
$CO_2$ efflux from the soil to be isolated from ambient atmospheric concentrations. A pump (NMP series micro diaphragm gas pump, rated for 2.5 liters/min of delivery, KNF Neuberger, Inc., Trenton, NJ) with a flow regulator (LI-COR Biosciences, Inc., Lincoln, NE) pulls the sample into an infrared gas analyzer (IRGA, Li-840A, LI-COR Biosciences, Inc., Lincoln, NE).

Each sample cycle consists of a 10 minute loop which includes a flushing period of 2 minutes to purge all existing gas in the sample lines. This is followed by a four minute incubation period, initiated by chamber closure, to incubate $CO_2$ and transport it to the Inferred Gas Analyzer (IRGA). The sampling loop ends with another two minute flushing period of the sample line.

4.2 Compression System

An AC or DC powered compressor is at the heart of the compression system at each site. The AC compressor (GAST 1HAB-11T-M100X, Idex Corporation Benton Harbor, MI) can generate gage pressure of up to 100 psi (690 kPa) and is equipped with a 2 gallon (7.6 L) air tank. The DC powered compressor (250-IG, industrial grade, Viair Corporation, Irvine, CA) is equipped with a motor capable of generating up to 150 psi (1034 kPa) in gage pressure. Both have an internal pressure switch, that turns on power at 85 psi and off at 105 psi. The DC powered unit was integrated into the sensing system at solar power operated sites.

Tests on the compression system showed that it consumes 9 Amps at 100% duty cycle and that minimal use over a daily cycle can be achieved by using a smaller air tank (1 gallon, 3.78 L), while deploying additional measures for automated power control and monitoring of air pressure. As a result, an external solid relay (40 Amps, Crydom H12D4840D, Custom Sensors and Technologies, San Diego, CA) and an external pressure transducer (PX309, Omega Engineering, Inc. Stamford, CT) were added to the system to closely monitor the compression system. This was the first step in isolating the power hungry compression system, and reducing the likelihood of total system failure due to excessive power draw and resulting loss of remote communication. Avoiding such a loss is critical to operating remote sensing stations and minimize downtime in data collection. For this reason we
isolated the power hungry system components (compressor and pump), along with their associated components onto a separate solar power system that is isolated from the soil sensors, data logger, and intergraded system telemetry.

4.3 Soil Moisture and Temperature Sensing

Each chamber location is paired with soil moisture sensors. We selected two types of sensors (5TM water content and temperature sensor, Decagon Devices., Inc. Pullman, WA; CS650 water content reflectometer, Campbell Scientific, Logan, UT), each with different physical characteristics for ease of installation at the depths selected for the study 5, 15, and 30 cm). A set of 8 to 10 sensors were tested for accurate measurements by inserting them in soil in the laboratory, testing them for accuracy and inconsistent behavior using a calibrated hand-held soil moisture sensor.

A soil pit, approximately 30 cm in width and 40 cm in depth pit, was dug next to each automated chamber (Figure 3(left)). The evacuated soil was carefully stored in order of its removal for later use as fill ((Figure 3(right))). The 5TM sensor (10 cm in height 3.2 cm width, prong height of 5 cm and a zone of influence extending 1cm beyond the tip of the prong, and 2 cm along the side of the prong) was inserted vertically into the top layer of soil on the side closest to the automated chamber of soil until the prongs were concealed (Figure 3(a)). This allows measurement of soil moisture and temperature indicative of the layer of soil commonly known as organic or O- horizon. Most of surficial organic matter and microbial activity resides in non-decomposed form in the O-horizon. The CS650 sensors (38.5 cm long and 6.3 cm wide, with prong height of 30 cm and a zone of influence extending 4.5 cm beyond the tip, and 7.5 cm along the side of the prong) was inserted horizontally into the soil at 15 cm and 30 cm depth, in layers of mineral soil. The evacuated soil was deposited back into the pit in order of its retrieval. Test measurements on the sensors were collected for 1-2 weeks, allowing the soil to settle and minimize disturbance, and the sensors to equilibrate to their surroundings.

4.4 Ancillary Measurements

Each sensing system is equipped with additional sensors to provide data of relevant environmental parameters and support the safe and continued operation of the system. Sensors measuring air and chamber temperature,
precipitation and wind speed, were added to the sensing system. One air temperature probe (107-L, Campbell Scientific, Logan, UT) housed in a radiation shield (41303-5A, 6-Plate Gill Solar Radiation Shield R.M. Young and Company, Traverse City, MI) was installed at each site. Each automated chamber is equipped with a temperature sensor, custom-built with a thermistor-type sensor (7002, thermistor, Campbell Scientific, Logan, UT).

A precipitation gage capable of measuring rain and frozen precipitation (52202-L30-LP30, Heated Rain and Snow Gage, R.M. Young and Company, Traverse City, MI) was added to select sites. The precipitation gage is protected from wind-induced measurement errors with a rain gage screen (260-953, Alter-type Rain Gage Wind Screen, 36 inch legs, NovaLynx Corporation, Grass Valley, CA). A wind monitor, capable of providing wind speed and direction (05103-10 Wind Monitor, R.M. Young and Company, Traverse City, MI) was installed at each site.

The safe operation of a remotely operated solar-powered sensing system requires close monitoring of system power. We used two independent solar-powered battery banks to divide up system components. One battery bank and associated solar panel array was equipped to adequately power the gas sampling and compression systems. To monitor power consumption we employed an external voltage measuring device (VDIV10:1, 10-to-1 Voltage Divider Terminal Input Module, Campbell Scientific, Logan, UT) to record the battery banks voltage. Voltage measurements and pressure in the compression system (used for chamber operation) are made twice hourly, prior to and after a sampling cycle. We used the in-built voltage reading capability of the data logger to read the voltage of the smaller battery bank used to power the soil sensors, datalogger, and telemetry system.

4.5 Datalogging and Telemetry

A data logger and associated hardware is used to control the system components and receive, store and transmit data. Due to multiple sensor inputs and a complex set of data acquisition tasks, we chose an off-the-shelf data logger (CR1000, Campbell Scientific, Inc. Logan, UT) with a variety of features (16 single ended analog inputs, 2 pulse counters, 8 digital ports, 2 communication and data storage ports, data storage expandable from 4mb to 4GB, compatibility with a variety of communication protocols and hardware), and an easy-to-use but powerful language (CRBasic) for system control and data collection.

Due to the distribution of sensing systems across the state, real-time remote data collection and system control capability is critical to the operation of the network. We reviewed several commercially available telemetry technologies including satellite, cellular phone and radio communication to determine which had the greatest ease of implementation, level of system control, and a manageable cost of installation and continued operation. Based on these factors a cellular phone based telemetry system was found to be most suited for implementation. We chose a cellular phone modem (Airlink RavenXTV, CDMA technology, Sierra Wireless, Carlsbad, CA), that is compatible with the datalogger, with an antenna (14201, 900 MHz 9 dB, Yagi antenna, or 14221, 900 MHz 3 dB, omnidirectional antenna, Campbell Scientific, Inc., Logan, UT, for sites with weak or strong cellular signal strength respectively). We used off-the-shelf software (Loggernet, Campbell Scientific, Logan UT) for system control and data collection. The modem connects to the data logger through its RS-232 port, and at 115200 Baud provides high data transfer rates, and lag-free system control allowing tasks such as automated data collection, program initiation, upload, termination, and other system troubleshooting tasks.
5 Design Strategy

5.1 Energy Budget

Two of the seven sites had preexisting access to line power. We installed line power at a third and these systems were implemented with no power constraint with a resulting continuous 24 hours a day operation. The operation of four sites was implemented with solar power. With the remote, forested nature of the sites, considerable effort was made in developing the power supply infrastructure and in optimizing the system design to enable uninterrupted operation.

The power consumption of a sensing system was estimated at 700 mA per sampling cycle with daily demand at remote sites of approximately 12.9 Amp-hours at duty cycle of 30% (i.e. the system is operating and drawing power only 30% of the day). With a plan to add additional sensors in the future, the system required a minimum supply of power of 800 mA per sampling cycle (100% duty cycle for the time when the automated chamber measurements and soil moisture and ancillary environmental measurements are in effect). The corresponding daily demand was estimated at 13.6 Amp-hours at 30% duty cycle, but in order to account for any uncertainties in the estimate the power supply infrastructure was designed for a daily demand of 15 Amp-hours. It was determined that the bulk of power consumption was attributable to operating the compression system and performing the SR measurements. This allowed the design of the power supply system with two independent inputs, one to power the compression system and the automated chamber and the other to operate data logger, soil moisture sensors and the remaining components of the sensing system.

5.2 Solar Power

Site characteristics such as topography, vegetation cover and type heavily influence the amount of sunlight available for power generation. Typical solar power infrastructure consists of photovoltaic cells that convert sunlight to electricity, charge controllers that regulate the electricity generated, and allow a bank of batteries to be safely charged.

Based on initial testing and an estimated average sunlight of 5 hours each day we developed a base design of the system that included photovoltaic cells with a total capacity of 150 W, and a battery bank capable of providing power for up to 10 days. This design was initially implemented at each site and tested for its performance and based on the response observed specific to local conditions, additional photovoltaic cell capacity and battery back-up was added. Final implementation of solar power installation include photovoltaic cells (AltE Poly 50 W or 80 W panels, AltE Store, Boxborough, MA) with a total capacity ranging from 150 W to 210 W, solar charge controllers (SG-4 Sunguard 4.5 Amp 12 Volt, or SunKeeper SK-12 12A, 12V PWM Charge Controllers, Morningstar Corporation, Newtown, PA) and batteries (Xtreme, Marine Deep Cycle, 125 Amp-hours, Batteries Plus, Hartland, WI ) with two batteries connected in parallel powering the datalogger and the soil moisture sensing system and five or six batteries connected in a similar way to power the compression and automated chamber control system. The installation of the photovoltaic panels was done selectively on posts, trees, or in one case a custom built tower in close proximity to the sensing system. The battery banks and charge controllers along with the system control and telemetry equipment were housed in an off-the-shelf shed (GS3000, 1.2 m³ volume, Suncast Corporation, Batavia, IL).

5.3 System Control

Code written in the CRBASIC programming language (Campbell Scientific Inc., Logan, UT) provides control of the individual tasks of the sensing system. A typical cycle of operations at a remote solar-powered site begins
with the data logger ensuring that the battery bank voltages exceed the voltage of safe operation (11.5V), and then proceeds to generate a random number to select a chamber for operation. Tasks before the operation of the chambers include the measurement of all soil temperature and moisture, air temperature and wind speed sensors. The system then checks the pressure level in the compression system and powers the compressor if it is below 80 PSI. Power supply to the compressor is terminated after two minutes or earlier if the compressor pressure reaches 105 PSI, and then the system proceeds to perform automated chamber control tasks. The sampling pump is turned on simultaneously with the relay controller activating the sample lines according to the chamber selected. The pump flushes the line for 2 minutes prior to the start of sampling cycle. After which chamber operation commences with the initiation of the IRGA to measure $CO_2$ concentrations every 3 seconds. The chamber lid is closed for 4 minutes after which the chamber lid is opened and the system lines are flushed for a further 2 minutes. $CO_2$ concentrations are measured and recorded for the 10 minute cycle, which once complete directs the system to go into a low power state and wait for the next cycle to commence.

6 Current Data

Sensing systems at select sites began operation in 2012, and over a period of 12 months the remaining sites were brought to operational capability for core and most ancillary measurements. Following the initial deployment we continued expansion of the sensing systems to include elements such as winter measurement of soil $CO_2$ efflux. While upgrading the systems they continued to operate and collect data with only minimal down time while being enhanced. An example of data collected is provided here to reveal the capability of such sensing systems.

Figure 4 provides a comparison of a year-long time series of soil moisture present in soil (up to a depth of 30 cm) between two sites. Volumetric water content measurements and in-situ soil properties (bulk density and porosity) were used to estimate the area-averaged volume of moisture available at a forested (SBM) and pasture site (BDF). There is more available moisture at the pasture site than the forested site. This can be explained by a combination of soil type and vegetation type differences. The pasture with its lack of large plants has lower water demand (evapotranspiration, ET). The presence of tree roots at the forested sites increases the effective porosity of soil that allows more water to infiltrate to deeper layers of soil and decrease water availability in the shallower layers of soil. During dry periods, a higher level of ET increases water demand at the forested site relative to the pasture site. This provides a close look at moisture drawdown during prolonged dry periods. Such data in combination with the knowledge of vegetation type can be used to determine the effects of prolonged and/or frequent dry periods on the health and productivity of forests.

7 Summary and Conclusions

We provided a detailed examination of the development of a soil processes monitoring network. The network was installed across seven sites representative of different land-uses in the state of New Hampshire. Each sensing system was designed to overcome the challenges posed by site topography, vegetation type and remote location along with excessive power requirements of the system components. This resulted in the development of a robust and optimized network of sensors built to provide uninterrupted data and help increase our understanding of the environment around us.
8 Acknowledgements

We acknowledge funding provided by National Science Foundation (Award # EPS-1101245) in the development of this work. Chris Cook provided valuable support in building many components of the system. Ruth Varner and Alix Contosta shared design information that helped us develop the automated chamber and gas sampling systems.

References


Challenges to using decentralized spatial algorithms in the field: The RISERnet geosensor network case study

Matt Duckham, Xu Zhong, Kevin Toohey
University of Melbourne, Australia

Abstract

Over recent years considerable research effort has focused on developing decentralized algorithms for highly distributed computing environments, such as wireless geosensor networks. There are several putative advantages of decentralization, including scalability, energy efficiency, and operational latency. However, decentralized algorithms today are primarily found in simulations and lab-based deployments, but rarely if ever in the field in true deployments. In this paper, we review the principles of and drivers behind decentralization. We then contrast these drivers with a recent field deployment of a large wireless geosensor network for monitoring environmental conditions relevant to wildfire hazard, called RISERnet. The comparison highlights the key areas of difference, where current technology and applications of wireless geosensor networks are not yet able to take advantage of decentralization.

1 Decentralized algorithms

A decentralized system is a special case of a distributed system where no single system component knows the entire system state [10]. Decentralized systems have a long history in computer science. Important fundamental advances have been made in decentralized algorithms, such as in efficient algorithms for leader election [12]. The design of decentralized spatial algorithms have a more recent history, but have grown into an active area of current research [11].

Decentralization is especially well-adapted to highly distributed systems, such as wireless sensor or geosensor networks. In a decentralized wireless geosensor network, computing happens in the network itself, with decentralized algorithms run in parallel on every node. There are four primary potential advantages of using decentralized algorithms, when compared with centralized alternatives [5]:

1. **Scalability**: As networks scale from tens to hundreds, thousands, or even millions of nodes, centralized architectures struggle to manage the increasing number and complexity of connections. Decentralized architectures are more scalable, allowing nodes to be easily added or removed, with each node executing the same procedures in parallel with its peers.

2. **Energy efficiency**: Decentralized algorithms enable in-network computing, where low-level data processing and filtering can occur in situ amongst groups of nearby node neighbors. In turn, this can lead to less information being communicated between nodes, improving the overall energy budget of the network.

3. **Operational latency**: Processing data in the network may avoid the need to communicate data from the network to a central sink for processing. For sensor-actuator networks, where data may be needed in the network to effect changes using actuators, reducing communication can in turn reduce the operational latency of the system.
4. Managing information overload: The volume of data generated even by relatively small wireless geosensor networks can be formidable, with each individual data item typically of almost no value. Enabling low-level collaborative processing and filtering of data in the network embeds intelligence in the network, reducing the potential for overwhelming applications and users with near-meaningless data.

In the domain of geographic information and wireless geosensor networks (GSNs), these advantages have fueled considerable research interest in the design and development of decentralized spatial algorithms. Examples include [8], who examined efficient spatial queries in the context of geosensor networks; and [13], who developed a decentralized sweep algorithm for GSNs, used as a building block for many higher-level algorithms. Many examples of other decentralized spatial algorithms, such as sweeps, can be found in [5].

Despite these advantages, it is fair to say that relatively few decentralized spatial algorithms have been tried and tested in the field. In the following section 2 we look at a specific example of a wireless geosensor network, called RISERnet. Despite the putative advantages of decentralization, discussed above, RISERnet currently does not make significant use of any decentralized algorithms. Hence, Section 3 examines more closely the reasons for this omission in the case of RISERnet, with lessons for GSN and decentralized spatial algorithms more generally.

2 RISERnet: A GSN in the field

The RISER project (resilient information systems for emergency response) is developing technologies capable of capturing, collating, and communicating timely and relevant information, even in the extreme and unexpected circumstances surrounding an emergency. As part of the RISER project, a GSN for monitoring environmental variables relevant to wildfires has been developed and deployed in the field. Wireless geosensor networks are recognized as an important tool for environmental monitoring for wildfire applications [1], with past work including both image-based monitoring of forests [6, 9] as well as fine-grained monitoring of environmental conditions relevant to wild fires [2].

The overall architecture of the RISER system is shown in Figure 1. This architecture includes three main components: the GSN itself (RISERnet); a stream-processing middle tier, including database archiving; and a real-time user interface.

![RISER system architecture](image-url)
2.1 RISERnet

RISERnet itself is a redeployable wireless sensor network for measuring environmental conditions pertinent to wildfire hazard and behavior. The network currently consists of 70 wireless sensor motes (Libelium Wasp-motes). Each geosensor node in the RISERnet network captures data from its on-board sensors a regular interval (configurable, but by default every 60 minutes). Nodes communicate real-time sensed data to one of four special gateway nodes (called Meshliums) via an XBee DigiMesh (2.4 GHz) multihop network. Gateway nodes forward aggregated data via a 3G WAN connection to the middle-tier.

Mesh networking  The multihop DigiMesh mesh network protocol eliminates the need for always-on coordinators. This reduces the frequency of battery changes in field. Furthermore, as the network does not rely on coordinators, the network is more resilient to single node and link failures. Although the line-in-sight communication range of the XBee module can be hundreds of meters, undergrowth and trees cause significant signal attenuation. In practice, the motes are on average about 80m from their nearest neighbor. Different network topologies have been adopted across the three different deployments where RISERnet has been used. The network has been deployed in a regular grid as well as an irregular deployment, to adapt to changes in vegetation density (motes were distributed more densely in the denser-vegetation, where signal attenuation is higher; in sparse-vegetation, the motes were positioned farther from each other).

Time synchronization  Power constraints exist in most wireless sensor networks deployments. Minimizing power consumption is critical for wireless sensor networks in the forest because a) higher signal attenuation leads to more frequent communication failures and retries; and b) solar panels are not usable in the forest. The most efficient solution for RISERnet was found to be duty cycling, where the network hibernates between sampling cycles. However, in order to establish multihop communication, all the motes must wake up simultaneously. Thus, time synchronization is essential in RISERnet. As the spatial extent of the network is relatively small (less than one square kilometer), RISERnet uses a simple time synchronization protocol. At the beginning of a sampling cycle, the gateway broadcasts a beacon which contains the current system time of the gateway and the next wake-up time of the network. The motes then synchronize their clocks with the gateway and set the wake-up time to that contained in the beacon message. The sampling period of the networks can then be changed simply by adjusting the setting in the gateway, which can be achieved remotely from the middle-tier.

Deployments  The first RISERnet network was deployed in Olinda in Victoria, Australia (see Figure 2). Each mote was armed with a temperature and relative humidity sensor (SHT75), a soil moisture probe (Watermark), a weather station probe, and a solar radiation sensor (SQ-110). Across the area covered by the network, these sensors enable the monitoring of factors relevant to wildfire hazard, such as fuel moisture content (FMC), at fine spatial and temporal scales.

A significant feature of RISERnet is its extensibility: the network can easily be redeployed, and reconfigured with new sensors. For example, a new RISERnet network armed with two new types of sensors has recently been deployed in Powelltown, Victoria. The new sensors, originally designed to measure soil temperature (a PT1000 sensor) and soil moisture (a VH400 sensor), are used to monitor fuel conditions directly.

Indeed, redeployment is now simple enough for a third small RISERnet network to have been established in Anglesea, Victoria, by local primary school children, as part of a broader and longer-term fire education initiative.
involving the whole Anglesea community. With assistance from the RISER team, the deployment was planned and conducted by school children. The children also maintained the network, replacing batteries as necessary and fixing any problems with the sensors that occurred following deployment.

2.2 Middle-tier

The gateways of the RISERnet networks forward the data collected by the sensor motes to the middle-tier through a messaging protocol called MQTT \[7\] over a 3G mobile network. Such real-time data is more suited to online data management systems than offline systems. Stream processing systems are amongst the most familiar class of information systems that adopt an online approach to information processing. RISER uses IBM InfoSphere Streams, a commercial stream computing platform with a modular, component-based programming model. The RISERnet stream processing system forms a bridge between the sensor data streams emanating from the RISERnet network and all the uses of these streams. These uses include archiving in a traditional spatial database (MySQL); processing to generate interpolated, high-resolution maps of current conditions; and display of data in a real-time user interface.

2.3 Interface

The RISERview interface presents current and historical data from RISERnet. The RISERview interface shown in Figure 3 is built on top of a Node.js server using a Leaflet and D3.js based map interface. The interface includes a timeslider to allow access to both real-time and historical sensor data (Figure 3(a)) over the whole network, as well access to more detailed current and historical data from individual nodes (Figures 3(b) and (c)).

Figure 3: RISERview mobile phone interface
3 Comparison

The RISERnet system described above currently operates without any use of sophisticated decentralized algorithms, despite the RISER team possessing significant expertise in their design and implementation. An obvious question is to ask why this is, and whether this omission arises from a failure in the sensor network technology, in the decentralized algorithm research, or some combination. This section looks briefly at each of the potential advantages of decentralization in the context of the RISERnet sensor network.

3.1 Scalability

Scalability is perhaps the most important potential advantage of decentralization. Even though the RISERnet network is a modest 70 nodes in size, the network is significantly larger than previous GSN deployments in forest environments (e.g., compare with \cite{3,4,6,9,14}, all less than 35 node networks) and scalability remains a challenge for RISERnet. Although the network is designed to be redeployable, for example, it does still require significant human labor to physically uninstall, move, and reinstall the network, with every manual operation repeated 70 times over, at significantly different locations, and in remote and harsh natural environments. Even after installation, the network is vulnerable to animals, such as small mammals. For example, a damaged weather station probe and a damaged sensor cord are shown in Figures 4(a) and 4(b), respectively. Cables that run along the ground are housed inside a garden hose and are protected at the ends with tape and rubberized fabric.

![Figure 4: Examples of a damaged anemometer (a) and a damaged sensor cord (b).](image)

While decentralized algorithms might be able to offer only limited assistance with such physical scalability issues, a purely peer-to-peer approach does hold the potential to improve operating reliability and simplify the networking and (re)deployment process.

However, in practice such scalability advantages are difficult to realize. Most significantly, the reliability of the underlying mesh networking services of today’s sensor network technology is not high enough to support reliable decentralized algorithms. The XBee DigiMesh multihop wireless communication protocol automatically finds a route (if there is one) to relay a message to the gateway, even when node or link failures happen. Hence, in most cases data from most nodes can be relayed to the gateway despite the relatively high frequency of node and link failures that occur even in relatively simple RISERnet network topologies and small network diameters (of at most ten nodes) used in our deployments. However, decentralized algorithms require not only reliable multihop communication between nodes and the gateway, but between all nodes in the network. Such reliable peer-to-peer networking is a prerequisite of decentralized computing, and is not well supported by current networking technologies.

Second, decentralized algorithms remain challenging to code and debug. One example of an early protocol failure in RISERnet occurred when a gateway fault resulted in repeated and frequent “handshake” messages from motes. This in turn led to rapid depletion of the mote batteries, leading to a loss of data and necessitating the replacement of all the batteries in the network several weeks earlier than originally scheduled. The high
practical and labor costs of such errors tend to militate against using anything but the most basic in-network protocols.

Finally, although 70 nodes deployed in the field is a moderately large research network, RISERnet is not large enough to provide the levels of spatial detail required by many spatial algorithms. Hundreds or thousands of nodes in a single area may be needed to provide a fine-enough spatial granularity to discern complex spatial events, such as topological changes in regions. Today’s wireless sensor network technology is not yet at the level where such large networks are practical or affordable, and so the need for decentralized spatial algorithms is lessened.

### 3.2 Energy efficiency

Developing an effective energy budget that enables the network to continue operating for extended periods of time remains one of the most challenging aspects of deploying any wireless geosensor network today. Extensive work on RISERnet has led to a network that can operate without battery replacement for up to six weeks. Achieving this longevity has required a range of technical innovations, including the modification of nodes to enable external rechargeable battery packs to be used (Figure 5). The gateways do not support automatic hibernation and wake-up, and so customized power controllers needed to be developed to turn on and off the gateway automatically, making sure to shut down the gateway’s file system before powering it off.

As discussed above, duty cycling is used to ensure the network can be powered off for any times when the network is not capturing data. Given our application of monitoring environmental parameters relevant to wildfires, a duty cycle frequency of between 15 minutes to an hour was adequate to capture any salient changes to the environment. In the context of such duty cycling frequencies, the costs of data communication are relatively small components of the energy budget, when compared to the costs of powering up nodes and sensors and the communication overheads required for time synchronization. In this context, decentralized spatial algorithms have the potential to offer only marginal benefits to the overall energy budget.

### 3.3 Operational latency

The operational latency of RISERnet (the time between data being captured and being made available to an end user) is low compared with the RISERnet duty cycle frequency. While our application needs can be satisfied by sensing data every 15–60 minutes, using our stream-based architecture in Figure 1 the sensed data is available online within a few minutes of being captured. Since these operational latencies are significantly shorter than the duty cycle periodicity, there is little to be gained from achieving the lower operational latencies that might result from decentralized algorithms. It is conceivable that other applications (such as monitoring of active firefronts) might indeed benefit from operational latencies lower than a few minutes. However, in the context of ongoing environmental monitoring, a centralized architecture is typically adequate.

### 3.4 Information overload

Information overload is an important issue, even for wireless geosensor networks of modest size and sensing frequencies such as RISERnet. The data generated by only 70 nodes, sampling a few times an hour for three months (the current length of time RISERnet has been continually operating) can still give rise to information overload issues. Note that it is not the volume of data that poses the problem; such data volumes and frequent updates are not especially challenging on their own. Rather it is the volume of data given the value of each
individual data item. For example, knowing that node W08 measured a fuel temperature of 13.57°C at 13:41 on April 05 2015 is in isolation practically meaningless. Rather, it is only information about the upward trend of fuel temperature across a nearby region of nodes over a period of days, in combination with a coincident drop in fuel moisture, that together gives rise to meaningful knowledge (about potentially increased fire hazard). Thus, managing the potential for information overload concerns managing the volume of data in the context of the meaningfulness of individual data items.

In the case of RISERnet, decentralization certainly might assist with managing information overload, by identifying meaningful coordinated changes in the network. However, as decentralization is already of only marginal benefit (as a consequence of other scalability, energy, and latency considerations), the stream processing architecture used in RISERnet is also capable of managing the potential for information overload. Stream operators executed over the RISERnet data are able to identify salient and coordinated changes in sensed data. In short, decentralization is one mechanism for managing the potential for information overload, but not the only mechanism available to the GSN system architect.

4 Conclusions

This paper examined that gap that still exists between research into decentralized spatial algorithm design and practical wireless geosensor network deployments. With today’s technology, practical considerations often negate the potential advantages of decentralized algorithms. Specifically,

- The low reliability of truly peer-to-peer mesh networking combined with the low levels of spatial detail afforded by networks comprising dozens rather than thousands of nodes leads to difficulties realizing the potential for increased scalability of decentralization.

- The impact of duty cycling upon the energy budget of the network may dwarf the costs of communicating data to a central gateway, particularly in cases where relatively low-frequency updates are required, such as ongoing environmental monitoring.

- In applications, such as monitoring ongoing environmental changes, where operational latency is lower than the required sensing frequency, decentralization has the potential to offer only marginal benefits in terms of operational latency.

- While the potential for information overload remains an issue even for networks of modest size, decentralization is not the only approach capable of helping to manage large volumes of volatile, low-reliability data. Hence, in networks where decentralization is not required for one or more of the reasons above, other techniques such as stream processing can substitute well in identifying meaningful patterns in dense data.

In summary, while clear reasons for the interest in decentralization exist, today’s technology combined with some application requirements may not always require decentralization. However, as the technology improves (e.g., increased ease of embedded programming, lower hardware and deployment costs of sensor nodes, increased reliability of peer-to-peer mesh networking), the potential benefits of using decentralization may also increase. Further, with improvements in technology, it is to be expected that new applications with requirements for higher frequency monitoring will become more commonplace, also strengthening the case for decentralization. Thus, while decentralization is not necessarily the approach for now, it still seems plausible it may be a significant component of the approach of the future.
References


Real-time Sensor Data Streams

Silvia Nittel
Spatial Informatics, School of Computing and Information Science, University of Maine, USA

Abstract

Today’s observation streams of large numbers of geographically distributed sensors in geosensor networks arrive continuously at powerful servers, and users are interested in analyzing the sensor data streams in real-time. This article investigates challenges in data management that arise when massive real-time data streams become available and discusses different data management technologies for managing and analyzing real-time sensor data streams.

1 Introduction

Geosensor networks [16] consist of large sets of sensor nodes deployed in geographic space. A sensor node is a combination of a computational unit and one or more configurable sensing devices. The computational unit is programmable and runs user programs that encode the user’s data and analysis needs. These needs include controlling the sampling frequency and determining how to handle the sensed data, that is, how to analyze and/or store the information locally, discard it if nothing ‘interesting’ is detected, or send it to other nodes or the cloud.

Sensor nodes mostly use wireless communication to send observations to other nodes. One of the many variable elements of a geosensor network architecture is the communication network topology. Traditional geosensor networks use mesh topologies [3] since they are flexible, robust and scalable. In a mesh network, individual nodes are connected to each other and base stations through multihop routing. A base station is a powerful computational node that can buffer data and is often connected to the Internet. In practical cases, Internet connectivity can be costly and limited, particularly in remote areas. Therefore, multimesh networks offer a scalable, inexpensive alternative. In a multimesh network, individual nodes are connected to each other and one or more base stations via a fine-grained radio mesh network, while a second, coarser-grained mesh network connects base stations with each other and the cloud. In this way, geosensor networks can scale up to very large numbers of nodes.

Mesh topologies with data centric routing and in-network collaboration have received much attention in research. In real-world geosensor network applications, raw sensor observations are commonly streamed out of the geosensor network, and integrated and analyzed on a more powerful server outside of the network [17, 12]. This availability of potentially very large numbers of sensor nodes – streaming observations in high frequency and in real-time – has led to novel data management and data analysis challenges. Users are interested in analysis of live streams compared to historic data, dealing with ‘big spatial data’ effectively, and real-time sensor stream analytics. Often, the question of appropriate data management support arises, and the recent explosion of new NoSQL systems (‘Not Only SQL’) in the database world does not make the selection easier.

In the past, SQL-like interfaces made it convenient to express data aggregation and subsetting tasks as queries, and delegated efficient execution to the database system. Today, the ‘system’ behind an SQL interface can come in different flavors, such as Hadoop-based implementations for large batch jobs over massive data sets, or real-time streaming engines working alongside traditional systems based on relational technology.
In this paper, I will discuss some of the data management challenges that arise when dealing with real-time streaming of larger numbers of sensors. The remainder of this article is organized as follows: in Section 2, sensor data streams are described in more detail. Section 3 addresses general challenges that arise when dealing with real-time sensor data streams. Section 4 discusses several state of the art data management technologies such as Hadoop-GIS, spatio-temporal database systems, and data stream engines with regard to their usability for different data management problems. Finally, conclusions are drawn and recommendations are given in Section 5.

2 Real-time Sensor Data Streams

A *sensor data stream* is a time series of sensor measurements \( m_{s_j} = < t_i, l_{s_j}, v_1, v_2, ..., v_n > \) generated by a sensor node \( s_j \), based on one or more of its attached sensors. To interpret a sensed value \( v_i \) correctly, we need the timestamp \( t_i \) and location \( l_{s_j} \) of the observation (and potentially, other information such as a sensor device identifier, sensor noise level, etc.). In this article, we consider time, location and sensed value as the minimum components of a sensor measurement. For practical purposes, not every sensor node is actually able to determine its own location using GPS; in this case, we assume the node is stationary and is initialized with its location during installation. The location of a sensor might also be derived indirectly from well-known locations of other individual sensors based on the sensor identifier at the server, and added to the sensor tuple on its arrival. More formally, an attribute \( a_i = (n_i, D_i) \) is a pair \( (n_i, D_i) \) where \( n_i \) is the attribute name (e.g. NO\(_2\)) and \( D_i \) is a value domain for the attribute (or a data type, e.g. floating point). A spatio-temporal relation \( R_{ST} \) is a finite subset of the Cartesian Product of the respective domains \( D_i \) of \( R_{ST} \)'s attributes, that is \( R_{ST} \subseteq D_{a_1} \times D_{a_2} \times ... \times D_{a_n} \) with the condition that at least one domain \( D_i \) is a spatial domain and a least one domain \( D_j \) is a temporal domain.

Informally, a sensor data *stream* is a continuously updating spatio-temporal relation \( R_{ST} \) with append-only tuples created by single sensor, i.e. the time series of updates of a single sensor. Alternatively, a sensor data stream can be seen as the continuously updating, spatio-temporal relation with append-only tuples of all sensors \( s_i \) in a geosensor network \( G \). This means that all sensors have the same schema, i.e. list of attributes. In the context of geosensor networks, viewing a sensor data stream as all updates of geographically and thematically related sensors is powerful, since we can reason about spatio-temporal events represented by the stream of observations. Thus, a *sensor data stream* \( S_s \) is a possibly infinite multiset \( S_s \) of time-stamped spatial tuples \( (t_i, s) \), where the spatial tuples belongs to the schema of \( S_s \) and \( t_i \in T \) indicates that the spatial tuple \( s \) arrived at \( t_i \).

3 Challenges

3.1 So, what is really new about sensor data streams?

One can ask the question “So, what is really new about sensor data streams?” Representing the updates of an individual sensor as a time series of data records has long been common practice in geographic and scientific applications. A traditional time series of spatial data tuples and a sensor data stream are similar if we look at the data structure: both consist of collections of spatio-temporal records, but spatio-temporal relations are *finite* sets of records while sensor data streams are possibly *infinite* multisets (there may be duplicates). The difference between relations and streams is mostly due to the data delivery and processing techniques. Because of inexpensive platforms, the geographic density of deployed sensors is much higher today [17, 12]. Secondly, sensor nodes use wireless communication so that observations are *available* for analysis in near real-time. Third, the *sensing frequency* per sensor has also drastically increased: samples every few seconds are common. For applications such as volcano monitoring, the frequency can increase to several thousand measurements per second. Characterizing sensor streams as a collection of spatially dense, high-frequency sensor observations in
near real-time nevertheless still leaves room for interpretation. What does “spatially dense” and “high frequency”
mean? In today’s pilot applications, we find deployments of sensors with numbers varying between 200 sensors
[15] to 20,000 sensors for a city-wide deployment [17], and sample rates vary between every 2 seconds to every
10 minutes. These numbers are bound to increase significantly over the next 5-10 years, with large sensor
network applications becoming more pervasive. One can expect that these advances in technology will enable
and inspire scientists to rethink how best to measure and sample phenomena that they are interested in, in order
to collect more data and reduce uncertainty.

3.2 User Requirements
Using spatially and temporally fine-grained geosensor networks with real-time sensor streams raises several
novel, interdisciplinary questions in different scientific and engineering areas such as electrical and chemical
engineering, computer science, spatial information science and environmental sciences:

- **Geosensor network design**: Geosensor network design will become a challenging question as sensors
  become cheaper and more abundant. In this case, geosensor network design is driven by the questions
  of how many sensors are optimal, and at which precise locations to deploy the sensors to get the best
  representation of the phenomenon and/or events of interest? Other considerations include which sensor
types to use, and analysis of the tradeoffs between accuracy, compatibility, price, and energy usage.

- **Real-time Sensor Stream Management**: Once a geosensor network is designed, built, programmed,
deployed, and tested the question becomes how to deal with the live-streamed data? One could store
the streams as files, assuming sufficient disk storage is available, and analyze the data later. However,
data management is not quite that simple if a user considers real-time data analysis needs. Choosing an
adequate data management tool will simplify the task of efficient and convenient real-time data analysis.

- **Spatio-temporal Data Analysis**: Typically, analyzing sensor data streams is a user’s primary interest.
Existing statistical methods remain unchanged with increasing data samples, however, they might require
new algorithms and implementations to handle much larger data sets and to deliver real-time results. Due
to the nature of the sensor web, geosensor networks can be composed of streams from different devices,
often under non-standard circumstances; this fact introduces varying accuracy and bias which needs to
be accounted for during analysis. Additionally, the focus of analysis will most likely to shift to spatio-
temporal analysis instead of mostly spatial analysis over snapshots of spatial data. Geovisual analytics
will play a more important role in analysis [6].

3.3 Data Management Challenges
With the availability of real-time sensor data streams, today’s analysis methods of spatio-temporal data are
not necessarily directly affected. The onslaught of sensor data streams, however, will contribute to the “big
spatial data” problem, but real-time analysis itself might not be necessary for many applications. For example, a
weather forecasting model will take recent data into account, and real-time forecast changes are not critical. On
the other hand, a public transportation monitoring system needs real-time analysis and response (e.g. identifying
accidents, traffic jams and rerouting etc.).

Novel challenges do arise with real-time analysis of sensor data streams. This type of analysis is performed
over a ‘window’ of the most recent data stream, and data analysis has to keep up with newly arriving data.
Analysis on raw sensor data streams includes looking for patterns in the raw data, cross-correlating raw streams
with other sensor streams, historic data and/or model predictions, and aggregating and summarizing raw sensor
data. Ideally, data management systems can provide convenient support for these tasks. We derive three main
requirements towards data management support for real-time sensor data streams.
• **Heavy lifting for raw sensor streams analysis**: A data management system should be capable of providing efficient support for real-time queries over very large amounts of sensor data streams and be able to keep up with incoming data.

• **Convenient integration with traditional data**: Since understanding, analyzing and interpreting current data is often performed through correlation with historic and/or model data, data management tools should provide seamless data representation and query capabilities between real-time sensor streams and historic and model data.

• **Data model support for continuity of time**: With frequently updating sensor data streams, data model support and query languages for the dimension of *continuous time* as well as spatio-temporal concepts become more critical. Both time as well as spatio-temporal concepts require formal foundations.

### 4 Data Management Support for Real-time Sensor Streams

In the following, we investigate the suitability of current data management technology for supporting analysis of real-time sensor data streams. We introduce categories of current data management approaches, and discuss data characteristics as well as suitable data management options.

#### 4.1 Hadoop-GIS

Today, much analysis of spatial data is performed using commercial or open source geographic information systems (GIS) tools such as R or Matlab. These tools are rich in libraries for geo-statistical analysis. GIS typically use files for data storage, and the task of subsetting data in or between files is up to the user. Stream data can be added as new files, but additional support might be necessary to speed up computation for large data sets. Popular tools are often based on Hadoop [9].

**Definition**: The Map/Reduce paradigm was introduced with Google’s technology to provide fast, scalable batch processing for tremendously large data sets like the index behind the Google search engine. The Map/Reduce technique allows seamless distribution and parallelization of code execution and data partitioning across many machines or the cloud. Open source implementations of Map/Reduce are e.g. Hadoop. In recent years, some GIS functionality has been rewritten using the Map/Reduce paradigm [9] using Hadoop-like systems for batch job processing. Thus, computing very large spatial data sets is sped up significantly [1, 2, 10, 5].

**Data characteristics**: Hadoop/GIS tools are useful for spatial analysis of previously collected, stored and potentially very large data sets, using any type of time intervals and spatial regions with regard to data. Data is stored in files and simply structured. This type of system does not contain out-of-the box functionality to deal with data streams characteristics, but novel architectures that seamlessly integrate batch job processing with the processing real-time data [13] are under development.

**Systems**: Hadoop-GIS [2], Spark R [5], ESRI tools for Hadoop [11], GeoMesa [10].

#### 4.2 Spatial and Spatio-Temporal Database Systems

Database systems (DBS) are convenient data management and declarative querying tools for large data sets. In this context, we define DBS as data management systems that are based on SQL and relational DBS technology (as opposed to SQL implement on top of Hadoop tools, also know as New SQL). In which scenarios are DBS appropriate to manage real-time sensor data streams?

**Definition**: *Spatial DBS* (s-DBS) are widely used today, both as open source and commercial systems. They have extensions (data types) to describe spatial data such as points, line string, polygons, networks and coverages and built-in spatial query support to make spatial search and simple spatial analysis convenient and
efficient. A spatio-temporal DBS (st-DBS) supports database concepts for both space and time information. It typically includes spatial types (as described above), temporal types (e.g. timestamp, interval, etc.) and potentially spatio-temporal types (e.g. a moving point or a moving region). Most of these s-DBS and st-DBS are built using relational databases systems. For higher dimensional data, array databases are available (e.g. SciDB).

Data characteristics: S-DBS are designed for transactional and stored data. New data records (inserts) are written to disk, and are immediately available for query and analysis. S-DBS are tuned towards very large datasets and high update rates (ca. 500 new tuplesinserts per second). Therefore, s-DBS and st-DBS are promising candidates for applications with sensor data streams that do not generate more than 500 new tuples per second across all sensors. Real-time data can seamlessly be integrated with historic stored sensor data, and queried on-the-fly. However, most s-DBS or st-DBS do not support continuous queries and it is the user’s responsibility to repeatedly restart a query. While s-DBS and st-DBS keep up with storing sensor data streams within certain update bounds, more complex analytical processing might not be executed in a timely manner for real-time analytics, since conventional algorithms heavily rely on disk based access and do not scale well to very large data sets [12, 19].

Systems: Oracle Spatial, IBM DB2, PostGIS, MySQL and others.

4.3 Data Stream Engines

Data stream engines (DSE) have been built as tools specifically to support high-throughput querying of real-time data streams with real-time answers. Besides financial and web analytics, sensor networks have also been a driving factor for DSE development.

Definition: A DSE is a data management system for real-time analytics of continuous data streams. Some DSE are similar to DBS, that is, they offer a data model language and query language so that information needs can be expressed using declarative queries. Other DSE require a heavy amount of programming to compose queries. In contrast to DBS, a DSE reevaluates a query repeatedly over newly arriving data instead of just once as in a DBS. Since most DSE are data-driven, a continuous query produces new results as long as new data arrives at the system. DSE do not support transactions and all data processing is performed in main memory. They do provide build-in modules to automatically deal with data bursts, and adaptive resource management across queries. ‘Older’ data can be pushed to a conventional DBS for longer-term storage.

Data characteristics: DSE are excellent data management tools for real-time analysis of very large numbers of frequently updating streams in applications such as emergency management and manufacturing. Current systems have throughput rates of 1,000,000 updates per second. Real-time data can be co-analyzed with historic data; however, the ‘window’ of processing is relatively short, for example a window can be few seconds or minutes long, starting from the current time and going back in time for e.g. five minutes. DSE do not support querying any kind of stream intervals involving older data which has expired from the live stream and/or stored in another system.

Systems: TIBCO Streambase [18], Apache/Storm [7], Oracle CQL [20], Microsoft Streaminsight [4, 14], IBM Infosphere [8].

5 Conclusions

Today’s geosensor networks produce large numbers of real-time sensor data streams. Sensor data is streamed directly to the cloud or a server for analysis, and users are interested in real-time data analysis. However, such set-ups of continuously streaming sensors bring new data management challenges. In this article, I explored these challenges in more detail, and discussed several options for potential data management support. The first option is centered around Hadoop-based analysis tools, which combine traditional spatial analysis with very
efficient batch job execution using the Map/Reduce paradigm. However, this approach is less useful for real-time analytics. The second choice is traditional spatial and spatio-temporal DBS, which provide declarative data modeling and querying, seamless integration with historic sensor data, and work well for up to 500 updates per second and simple analytical tasks. Thus today, they work well if the data streaming load is capped at about 500 inserts per second. The third option are data stream engines, which have been specifically designed for high-throughput real-time data analysis, and are appropriate candidates for very large numbers of sensor data streams and more complex analytics, providing real-time answers and continuous queries. The need for combing data management support for both high performance batch processing and real-time processing capabilities has led to the investigation of novel hybrid architectures such as the lambda architecture [13]. However, these systems are still under development at this time.

References


Fusing Human and Technical Sensor Data: Concepts and Challenges

Bernd Resch$^{1,2,3}$ and Thomas Blaschke$^1$

$^1$Interfaculty Department of Geoinformatics, University of Salzburg, Austria
$^2$Heidelberg University, Germany
$^3$Center for Geographic Analysis, Harvard University, USA

Abstract

As geo-sensor webs have not grown as quickly as expected, new, alternative data sources have to be found for near real-time analysis in areas like emergency management, environmental monitoring, public health, or urban planning. This paper assesses the ability of human sensors, i.e., user-generated observations in a wide range of social networks, the mobile phone network, or micro-blogs, to complement geo-sensor networks. We clearly delineate the concepts of People as Sensors, Collective Sensing and Citizen Science. Furthermore, we point out current challenges in fusing data from technical and human sensors, and sketch future research areas in this field.

1 Introduction

The predicted rise of geo-sensor webs has not taken place as rapidly as estimated approximately a decade ago. One may argue that this impedes a variety of research efforts from being carried out due to lacking near real-time base data. In contrast, we are currently witnessing the rapid emergence of user-generated data in a wide range of social networks, the mobile phone network, or micro-blogs. These human-generated data can potentially complement sensor measurements to a large degree, not by calibrated well interpretable measurements, but by subjective observations or human-generated measurements.

Current literature in the area of user-centred sensing oftentimes mixes up different approaches how data are generated, used and analysed. This paper distinguishes three concepts according to [10]. “1.) People as Sensors defines a measurement model, in which measurements are not only taken by calibrated hardware sensors, but in which also humans can contribute their subjective ‘measurements’ such as their individual sensations, current perceptions or personal observations. 2.) Collective Sensing tries not to exploit a single persons measurements and data, but analyses aggregated anonymised data coming from collective sources, such as Twitter, Flickr or the mobile phone network. 3.) Citizen Science stands for a human-based approach to science where citizens contribute semi-expert knowledge to specific research topics.”

This paper discusses particular challenges in fusing data from human and technical sensors (s. Figure 1), comprising standardisation (on data, service and method levels), data assimilation (resolution, aggregation, etc.), multi-dimensionality in the data, combination of methods from geoinformatics and computational linguistics (to extract information from user-generated data), quality assurance (both for technical and human sensors), and the consideration of privacy issues (data ownership, storage, optimum aggregation levels, etc.), and last but not least the fusion of user-generated data with remote sensing data.
Concepts: People as Sensors, Collective Sensing and Citizen Science

Ubiquitous sensor networks can assist in decision-making in near real-time in a broad range of application areas such as public safety, traffic management, environmental monitoring or in public health [13]. Yet, analysing and monitoring our surroundings in near real-time is still a major challenge due to sparsely available data sources [10]. As a result from this shortcoming, coupled with the fast rise of mobile phones, a number of researchers have started to investigate alternative methods for generating real-time data relevant for decision-making processes. Recent efforts have been taken by OpenSignal [8], On Line Disaster Response Community [6], CenceMe [7] or Near Future Laboratory [4]. In scientific literature, we see a number of human-centred sensing approaches that can be summarised under three main concepts: People as Sensors, Collective Sensing and Citizen Science [10]. This sub-section presents a clear disambiguation between these concepts.

“People as Sensors” defines a sensing model, in which measurements are not only taken by calibrated hardware sensors, but in which also humans can contribute their subjective measurements such as their individual sensations, current perceptions or personal observations [12]. Like this, people act as non-technical sensors with contextual intelligence and comprehensive knowledge. Measurements are not created absolutely reproducibly by calibrated sensors, but through personal and subjective observations. Such observations could be air quality impressions, street damages, weather observations, or statements on public safety, submitted via dedicated mobile or web applications. A vibrant real-world example is WAZE [14], a smartphone app allowing people to send their personal traffic reports, which are directly used in other persons routing requests. These human sensors can thus complement—or in some cases even replace—specialised and expensive sensor networks. Throughout recent literature, the term “People as Sensors” is used interchangeably with “Citizens as Sensors” [5] or “Humans as Sensors” [3].

A concept related to People as Sensors is Participatory Sensing, in which a number of persons with a common goal in a geographically limited area contribute geo-referenced data via their end user devices such as smartphones [2]. From this definition it is evident that the term Participatory Sensing is highly similar to People as Sensors, but its definition is a little more restricted in terms of input devices, data acquisition and information processing.

Furthermore, we are currently witnessing a fast rise of Collective Sensing approaches. In contrast to People as Sensors, Collective Sensing is an infrastructure-based approach, which tries to leverage existing Information and Communications Technology (ICT) networks to generate contextual information. This methodology tries not to exploit a single persons measurements and data. Collective Sensing analyses aggregated and anonymised
data coming from collective networks, such as Flickr, Twitter, Foursquare or the mobile phone network. Like this, we can gain a coarse picture of the situation in our environment without involving personal data of individual persons.

Finally, the term **Citizen Science** plays a key role in the context of People as Sensors. Citizen Science basically states that “through the use of sensors paired with personal mobile phones, everyday people are invited to participate in collecting and sharing measurements of their everyday environment that matter to them” [9]. An example for promoting the Citizen Science concept is the Citizen Science–Community Involvement Today and in the Future grant program by the US Environmental Protection Agency (EPA). This program aims to encourage individuals and community groups in New York City to collect information on air and water pollution in their communities, and seek solutions to environmental and public health problems.

Table 2 [10] summarises the comparison of the discussed concepts of People as Sensors, Collective Sensing and Citizen Science according to the following criteria.

- **Voluntary/Involuntary**: whether contributing people voluntarily (dedicatedly) share their data for further (geo-spatial) analysis or decision-making
- **Content**: type of data, which are contributed
- **A Priori Knowledge**: required knowledge of the user
- **Contextual Data**: whether the contributed data contain contextual intelligence, for instance a person’s local knowledge
- **Reliability**: quality level of the generated data and contributors’ trustworthiness
- **Analysed Datasets**: whether single (individual) datasets are analyzed or spatially and temporally aggregated (anonymised) data are used
- **Specific Infrastructure**: whether additional dedicated infrastructure is necessary to collect data

<table>
<thead>
<tr>
<th></th>
<th>People as Sensors</th>
<th>Collective Sensing</th>
<th>Citizen Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voluntary/Involuntary</td>
<td>Voluntary</td>
<td>Involuntary</td>
<td>Voluntary</td>
</tr>
<tr>
<td>Content</td>
<td>Layman Observations</td>
<td>Raw geo-data (images, tags, ...)</td>
<td>Semi-professional Observations</td>
</tr>
<tr>
<td>A Priori Knowledge</td>
<td>Medium</td>
<td>Low/None</td>
<td>High</td>
</tr>
<tr>
<td>Contextual Data</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Reliability</td>
<td>Medium</td>
<td>Mediocre</td>
<td>Good</td>
</tr>
<tr>
<td>Analysed Datasets</td>
<td>Individual</td>
<td>Aggregated</td>
<td>Individual</td>
</tr>
<tr>
<td>Specific Infrastructure</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 1: Comparison of Human-Centric Sensing Concepts[10]

### 3 Challenges in Fusing Human and Technical Sensor Data

In bringing the measurement concepts laid out in section 2 to practical use for data analysis, visualisation and communication for decision support, the central issue is how to fuse data from technical and human sensors
coming from a variety of sources as illustrated in Figure 2. This section illustrates the most pressing challenges, ranging from standardisation, quality assurance and fusion of different sampling rates to the representativeness/reliability of user-generated data and data privacy and protection issues.

![Figure 2: Increasing Availability of Technical and Human Sensors.](image)

The first challenge in fusing human and technical sensor data is the pertaining lack of standardisation. **Standardisation on data, service and method** levels fosters interoperability as a basis for joint analyses of human and technical sensor data. Even though reasonably mature data and service standards exist for technical sensors, for instance, through the OGC Sensor Web Enablement initiative [1], the integration of human sensor data including measurement processes, uncertainties, context variables, etc. is not possible yet.

A further challenge is the fusion of data from sensors with **diverse sampling rates and spatial resolutions**. First, this applies to sensors measuring in different temporal intervals (e.g., one sensor measuring in 3s intervals and the other one measuring in 7s intervals). Intelligent methods have to be found to combining these data apart from the least common multiple, which could be achieved by interpolation algorithms based on dynamic regression models depending on the sensors’ characteristics. Another dimension of complexity is added when attempting to fuse in-situ sensor data with remote sensing data regarding temporal measurement intervals, which induces additional challenges through the combination of point data with raster data.

Another challenge that is particularly related to analysing data from social media channels. Here, most previous geo-analysis methods focused on finding spatial patterns, neglecting the **multi-dimensional nature of social media data**. Thus, new methods need to be developed across the disciplines of GIScience, temporal analysis, computational linguistics, computer science, and others to fully leverage the potential of social media data. This also includes the correlation of social media data with “People as Sensors” observations and technical data to maximise information density. One example of such an integrated approach is demonstrated in [11].

A highly complex, but overly important issue is **quality assurance in technical and human sensor data**. For technical sensors, a number of methods exist (Kalman filtering, threshold detection, dynamic tolerance levels, comparison with neighbouring stations, flatline checks, spatial regression or comparison with modelled data) [15] as the properties of the measurement procedure are known. However for human sensor data, traditional
geo-data quality parameters as defined by the International Organization for Standardization (ISO) including accuracy, position accuracy, completeness, consistency, lineage, etc. need to be extended by new parameters such as expertise of the human sensor, spatial and temporal plausibility, or up-to-dateness. Even though these parameters might be rather simple to be defined, their actual assessment is highly difficult due to the unknown measurement process characteristics in human sensors.

The different nature of user-generated data results in differences in terms of semantic expressiveness of mobile phone data: The user-generated mobile network traffic represents a relatively large proportion of the population across social classes. However, these data are typically lacking content. For instance, the number of text messages sent or received might be logged, rather than the text itself; or the number and duration of voice calls might be logged, rather than the topic of the talk itself. This is in strong contrast to social media data and VGI, which are typically generated by a rather specific sub-group of the population, and explicitly contain content of some semantic value.

A connected methodological issue in the field of semantics is representativeness in VGI. This has to be tackled by a combined bottom-up/top-down approach. In bottom-up approaches, user groups and communities define their own semantic objects and interrelations between these in separate taxonomies. In contrast, top-down approaches try to define semantic rules and ontological relations as generically as possible—mostly before actual applications exist and decoupled from real-world use cases. Only the combination of those approaches can result in trans-domain semantic models, which are linked via object relations.

The requirement of high-quality information seems to be self-evident, but has not been tackled thoroughly for real-time geo-sensor networks and People as Sensors based approaches. The subjectivity of human “measurements” naturally raises the question of trustworthiness of these data in terms of data quality. As discussed above, this results in uncertainty in the observation data. Thus, automated quality assurance mechanisms need to be developed for uncertainty estimation, dynamic error detection, correction and prevention. Different approaches are in development, e.g., Complex Event Processing (CEP) for error detection, standardisation efforts for representing uncertainty in sensor data (e.g., Uncertainty Markup Language—UncertML [16], or proprietary profiles to define validity ranges for particular observations. Such issues need to be solved in order to ensure reliability of both technical and human sensor data.

Another pressing question is: how can we preserve people’s privacy when dealing with user-generated data and information, and partly sensitive personal data, in the context of mobile phones as ubiquitous in situ geo-sensors? In terms of privacy, the claim might arise that we need to be aware of our personal and private data before we share them. This also raises the need to discuss the concept of U-VGI, i.e. Un-Volunteered Geographic Information, in contrast to VGI [10]. For instance, Collective Sensing approaches exploit anonymised data from digital networks (e.g. by deducing crowd movements from traffic distribution in the cell phone network) even though people have not intended to share their data in this way.

As mobile phone data and human sensor data are individual oftentimes sensitive, legal frameworks have to be developed on national, trans-national and global levels to protect those personal data. The largest limiting factor in this regard is the varying interpretation of ‘privacy’ in different parts of the world. For instance, privacy can be traded like an economic good by its owner in the USA, whereas it is protected by law in the European Union. This means that supra-national legislation bodies and initiatives are called upon to set up appropriate world-wide regulations.

This also includes the critical question of data ownership. Shall they be owned by data producers, i.e., the citizens or a mobile phone network operator? Or rather the institution that hosts a system in order to collect data? Or the data providers? Furthermore, if sensitive data is analysed to produce anonymised information layers, who is responsible if decisions that are based on this information are wrong due to lacking quality of the base data? In conclusion, the issues of privacy, data ownership, accessibility, integrity and liability have to be tackled thoroughly all at once and not separately from each other.

Finally, remote sensing (RS) data are a somewhat special case within the realm of human and technical sensor data fusion. RS data are typically well structured—and may therefore not fall under the ‘big data paradigm’
even though such data sets can exceed the Terrabyte dimension. RS may often provide data of good comparability and repeatability (i.e., repeated measures for the same variables) and clearly defined types and levels of resolution. Until recently, the level of resolution of remotely sensed data was often seen critical for the integration with in-situ data, particularly the types of data described herein. Different societal research questions require different levels of resolution in space and time, and perhaps also regarding the spectral resolution. While 2015 marks a new dimension in spatial resolutions of civilian, publicly accessible satellite remote sensing—WorldView-3 offers 30 cm panchromatic and 1,24m multispectral resolutions – the variety of needs in human sensing suggests that any data fusion methodology should be highly flexible. We may conclude that since for RS data all meta-data are plannable and known the gained evidence about the world is predominantly dependent on the point and time of the observation. We may epistemologically call this realism–some may call it positivism while assuming that objects exist independently from the observer.

4 Conclusion

Without any doubt, fusing human and technical sensor data includes a wide variety of technical and methodological challenges. It may, however, comprise major epistemological problems. When combing data from different sources gathered under different schemas and for different purposes, one faces the potential problems of fusing ‘apples and pears’. This short article widely follows a techno-positivist epistemology while assuming that an objective reality can be explored and determined through scientific methods–basically observation and testing which create verifiable knowledge about the world. Obviously, the limits to this approach are not only determined by our ability to measure precisely. The key question is if we are measuring the right, i.e., relevant variables to the underlying societal questions.

This article categorises some of the major challenges in this predominantly technical domain. The exception to the technical dominance of this field and the potential lynchpin for supporting societal relevant research could be the citizen science approach. However, being a young development many citizen science projects may be at risk to be regarded as an end in itself: it is often investigated whether or not citizen science works and examples where citizen science was or is the key to solve a real-world problem which could not be solved otherwise are rare.

A scientific challenge lies in the trivial question whether evidence about the real world depends upon the perspective of the observer: Two persons who view the same object may interpret it quite differently because of their different personal history and assumptions about the real world. While GIS systems are technically mature and so may sensor devices be, we believe that the two major concepts discussed herein, namely ‘people as sensors’ and ‘collective sensing’–in combination with citizen science methods–are still in their infancy and need methodological and epistemological foundations.

References


Formal Foundations of Sensor Network Applications

Jacob Beal\textsuperscript{1}, Mirko Viroli\textsuperscript{2}
\textsuperscript{1}Raytheon BBN Technologies, USA, \textsuperscript{2}Universita di Bologna, Italy

Abstract

One of the key features that distinguishes sensor networks from other networked applications is that their focus is generally not the sensors per se, but space-filling phenomena of the environment through which the sensors are deployed. Following the mathematical implications of this observation leads to a formal grounding of sensor network applications in a field calculus that describes sensing, modeling, and interpretation of space-filling phenomena directly in terms of operations on mathematical fields. This points to more flexible, scalable, and resilient approaches to sensor network applications, as well as simpler approaches to developing decentralized applications that can provide robust services in difficult operating environments such as natural disasters, mass events, and critical cyber-physical systems.

1 From Sensors to Fields

The advent of cheap computer networking has transformed sensing applications, allowing a dramatic increase in both the number of sensors deployed and the speed with which sensor information can be accessed. The models used for constructing networked sensing applications, however, are still largely encumbered with the assumptions of generic computer networking.

Under a generic networking model, the focus is on individual devices, each of which may be a source of unique data services or user requirements. In sensor networks, on the other hand, each individual device is more often simply one sample of a space-filling environmental phenomenon, such as the temperature of the air, the flood stage of a river system, or the distribution of an invasive species. The fact that these space-filling phenomena are sensed with particular sensors at particular locations is of relatively little concern: the value of the individual sensors is their ability to help model the phenomena of interest. Even the care taken to ensure good placement of sensors is indicative not of their importance, but of their unimportance, as the aim is generally to ensure that the placement of sensor provides the most representative view of the phenomenon.

We would likely be well-served, then, to engineer sensing applications using models that consider not the samples, but the continua that they represent. Formally, both space-filling phenomena and collections of sensor measurements can be formalized as mathematical fields, where a field is a function that maps each point in some region of space to a data value. Sensor applications may then be specified at an aggregate level, modular with respect to the details of how sensors are distributed and networked, in terms of the acquisition and manipulation of fields. For example, the collection of stream gauges making up the Iowa Flood Information System (IFIS) \cite{6} may be thought of as sampling a field of sonar ranging data, which is then compared point-wise with another field of baseline measurements to produce the desired field: the estimated current flood stage at every location along the river systems that are being monitored.

Following the implications of this mathematical formalization has potentially profound impacts on the development of sensor applications. Formalizing data as fields leads to a formal field-based model of computation, the field calculus (Section 2), which in turn can form the foundation for aggregate programming models that
implicitly provide sensor network applications with flexibility, scalability, and resilience (Section 3). This in turn can enable simple design of decentralized applications for providing robust services in difficult operating environments such as natural disasters, mass events, and critical cyber-physical systems (Section 4), opening up possibilities for new high-impact applications in sensor-driven services and cyber-physical systems (Section 5).

2 Field Calculus Systematizes Computation with Fields

The observation that many applications can be thought of in terms of space-filling fields has been widely shared, not only for sensor networks, but in a wide array of other domains as well, including swarm robotics, biological modeling, and pervasive computing. This has led to the development of a large variety of approaches to programming at an aggregate level, including some tailored for sensor networks such as as Regiment [12] and TinyDB [9]: a thorough survey may be found in [2], and another focused on approaches specifically for sensor networks in [11]. Most of these approaches, however, have been rather ad hoc and specialized around particular assumptions about network structure and goals, making them generally difficult to apply pragmatically.

The field calculus [14] provides a more solid mathematical grounding derived from the common concept of “field” shared by many of these approaches and the commonalities in how such fields are manipulated. Its aim is to provide a “core calculus” that captures the essence of programming with space-time fields while remaining small enough for mathematical analysis to be tractable—just a \( \lambda \)-calculus does for functional programming [5], \( \pi \)-calculus for parallel computation [10], and FJ for object-oriented programming [7].

In particular, field calculus computation use an “everything is a field” paradigm: variables, computed values, sensed values, values feeding actuators, intermediate results of computations—all of these are fields. So in a sensor/actuator network we have the field of temperature detected on the ground, the field of users as detected by GPS sensors on smartphones, the field of identifiers for the Bluetooth beacons spread in a building, the field of Boolean values to activate lights in a smart-city, the field of vectors to guide unmanned vehicles or drones, etc. A computation thus starts from input fields, which are sensed environmental values or constants, and ends up with output fields representing information, either simply representing a model of the phenomena or being further fed to drive actions of the system on the basis of that model. Such a computation is carried on by progressively composing inputs using five basic constructs: “built-in” functions involving no communication or memory (e.g., addition, cosine, square root, local sensing or actuation), communication by sharing state with neighbors (operator \( \text{nbr} \)), memory via a state update construct (operator \( \text{rep} \)), branching by restriction of field domains (operator \( \text{if} \)), and calls to user-defined functions that abstract such combinations of constructs.

Critically, field calculus is both universal and has aggregate-local equivalence. Universality means that it can approximate any function on either a discrete network or a continuous region of space to an arbitrary level of precision [4]. At the same time, the particular choice of operations means that we can freely shift between aggregate and per-device models of computation, meaning that computations specified in terms of fields can be automatically transformed into equivalent programs to be run on the individual devices of a sensor network. Together, these two properties mean that field calculus supplies a general mathematical foundation for both implementation and analysis of any arbitrary sensor network application.

Finally, this mathematical framework has been implemented in a practically usable form by its instantiation in Protelis [13], a Java-hosted implementation of field calculus that allows simple integration of field calculus programs with the full range of Java libraries. As a simple example, the following code shows a user-defined function: \text{gossipMin} computes the minimum value of the input that has been sensed at any time at any location, and \text{boundedGossipMin} bounds this computation to the only that portion of space where region
holds true:

```python
// Gossiping is achieved by repeated updates of variable v to the minimum observed either locally or across neighbours
def gossipMin(value) {
    rep(v <- value) {
        min (value, minHoodPlusSelf(nbr(v)))
    }
}

// Branching construct if restricts the domain of each branch to a subspace, with no communication between subspaces
def boundedGossipMin(value, region) {
    if(region) {
        gossipMin(value)
    } else {
        null
    }
}
```

3 From Field Calculus to Practical Aggregate Programming

While field calculus is universal, it is also too low level for practical use in building complex distributed services. Mostly, this is due to the fact that any field computation that involves medium- to long-range propagation of information needs to suitably combine three different constructs (as shown in the gossipMin example): construct `nbr` to propagate information across a single step away, an built-in accumulation function (like `minHood`) to reconcile different values coming from different neighbors, and `rep` to chain information from step to step across a longer distance. Any non-trivial algorithm of collective behaviour, then, tends to require multiple stacked levels of such propagations, and can quickly become quite complex to build, read, debug, and maintain. This should, however, be seen as a flaw of field calculus itself: rather, it is the common characteristic of all core calculi, since their goal is not ease of programmability but the smallest set of ingredients necessary to support expressiveness in principle, as such terseness is vital for mathematical analysis of the properties of the calculus.

Instead, core calculi are generally rendered usable by using them to construct code libraries that raise the level of abstraction to a much more natural programming interface. For field calculus, we thus begin to raise the level of abstraction by identifying a collection of general “building block” operators for constructing resilient coordination applications, making the construction of increasingly complex services more practical. Each of these building blocks captures a family of frequently used strategies for achieving flexible and resilient decentralized behavior, hiding the complexity of using the low-level constructs of field calculus. Moreover, by using these building blocks exclusively, one can guarantee that the resulting system has some formally proven resilience properties [3].

Raising the level of abstraction yet further, on top of the building blocks one can stack additional layers of libraries (as shown in Figure 1), moving from general-purpose elements of self-organisation up to more specific mechanisms of collective adaptive behavior, and ultimately to sensor or sensor/actuator applications.

The critical layer of building blocks is formed by three new general operators `G`, `C` and `T`, along with field calculus’ `if` and built-ins, which behave as follows:

- **G** is a “spreading” operation generalizing distance measurement, broadcast, and projection, which takes as input four fields: `source` (a Boolean indicator field), `init` (initial values for the output field), `metric` (a function providing maps associating a distance to each neighbor), and `accumulate` (a binary function over values. It may be thought of as executing two tasks: it computes a field of shortest-path distances from the `source` region according to the supplied function `metric`, then propagates values along the gradient of the distance field away from source, beginning with value `initial` and accumulating along the gradient with `accumulate`. As an example, if `metric` is physical distance, `initial` is 0, and `accumulate` is addition over floats, the `G` creates a field mapping each device to its shortest distance to a source [8, 1].

- **C** is complementary to `G`, it accumulates information down to the gradient of a supplied `potential` field. It takes as input four fields: `potential` (a numerical field), `accumulate` (a binary function
over values), local (values to be accumulated), and null (an idempotent value for accumulate). Beginning with an idempotent null, at each device the local value is combined with “uphill” values on the potential field using a commutative and associative function accumulate. For instance, if potential is exactly a distance gradient computing with $G$ in a given region $R$, accumulate is addition over floats, and null is 0, then $C$ collects the sum of values of local in region $R$.

- $T$ deals with time, whereas $G$ and $C$ deal with space—since time is one-dimensional, it carried out the functions of both spreading and collecting. It takes as input three fields: initial (initial values for the resulting field), zero (corresponding final values), and decay (a unary operator over values). Starting with initial at each node, that value gets decreased by function decay until eventually reaching zero value. It is hence a flexible count-down toward zero, where the rate of the count-down may change over time, and on top of which any local state evolution can be implemented. For instance, if initial is a pair of a value $v$ and a timeout $t$, zeros is a pair of an unknown value null and 0, and decay takes a pair to remove the elapsed time since previous computation from second component of the pair, and turning the first component to null if the first reached 0, then $T$ implements a limited-time memory of $v$.

It can be shown [3] that when properly implemented, these building blocks, plus if and built-in operators, enjoy the following properties, which are then naturally inherited by APIs and applications built on top of them: stabilisation: if the input fields eventually stabilise to a fixed state, the same happens for the output field; resilience: if some messages get lost during system evolution, or some node temporarily fails, this will not affect the final result; adaptability: if input fields or network topology changes, the output field automatically adapts and changes its shape accordingly; and scalability: the performance of output field evolution is not affected by increase of the number of devices, provided that computation frequency and range maintain equivalent ratios.

On top of such building blocks, it is then possible to build libraries for addressing the various concerns of decentralized sensor network applications, including libraries for state evolution (e.g., timers, timed memory, integrators over time), distributed action (e.g., gradients, path forecasting, network partitions, broadcasts), distributed observation (e.g., summary, average, integral over regions), and more generally, collective adaptive behaviour (e.g., distributed situation recognition, leader election, collective choice, anticipative adaptation).
40

Figure 2: “Building block” operators for distributed services: $G$ spreads information from some sources outwards, progressively computing information en-route, $C$ aggregates information in a given region by progressively transmitting it to a sink area, $T$ aggregates over time by evolving a state variable until a fixpoint is reached, and $\text{if}$ performs a distributed branch, partitioning the domain into non-interacting subspaces in which different computations carry on.

As an example, in [3] it is shown how a specification of only around 20 lines of code, using a library API based on such building blocks, defines an entire decentralized application for crowd sensing at mass events via opportunistic interactions between personal devices: operator $G$ (via more user-friendly library functions) is used to sparsely partition the space into clusters of a characteristic size, operators $C$ and $T$ are used to model the crowd size and density within each cluster, and then $G$ applied again to warn of dangerously crowded areas, suggest directions of dispersal, and provide navigation services for circumventing dense crowds.

4 Applications of Aggregate Programming

Shifting from foundations to applications, the areas in which field-based computational models are likely to have the most impact are those in which decentralized data processing is valuable. Decentralized data processing is much more complex than centralized, cloud-based processing, however, and so is only likely to be embraced when there is a strong need for it. In practice, the rapid expansion of civil wireless infrastructure means that in many cases, there is no need for decentralized data processing: building WiFi, fast data over phone cellular networks, and cheap satellite communication via systems like Iridium means that in many applications, sensors can simply exfiltrate data directly to cloud facilities and remove any need for considering the additional complexity of decentralized processing.

In a number of critical application areas, however, the demand for communication badly outstrips the expected available resources, and decentralized data processing makes sense. The three main cases in which this applies are:

- **Wireless infrastructure is inherently lacking:** In some scenarios, a strong wireless infrastructure simply cannot be assumed to be available, such as natural disasters (in which the infrastructure may be damaged) or underwater systems (in which high-speed communication simply is not available). There are also scenarios such as military operations where wireless infrastructure exists, but must not be relied upon.

- **Extremely high density of information:** In other scenarios, wireless infrastructure exists and may be accessible, but the density of sensors or their rate of information generation is too high to be supported. Examples include mass crowd events such as marathons and street festivals, where the number of attendees overwhelms the cell phone infrastructure, or emerging sensor-embedded materials, such as smart fabrics or adaptable wing surfaces, in which there may be a very large number of sensors taking data at high rates.
- **Tight sensor-actuator loop:** The third main case is scenarios where on average the network may be able to sustain the communication, but safety-critical actions need to be taken in a timely fashion on the basis of the sensed data. Examples include UAVs and urban traffic management. In this case, the system cannot tolerate even occasional network slowdowns or disconnection, and at least some processing must be decentralized for the sake of safety.

In all of these types of scenarios, field-based computation models can be valuable because they separate the concerns of managing a resilient networking and adaptation to changes in the set of sensors from the more global application-level concerns, making many aspects of resilience and scalability implicit and thereby greatly simplifying the engineering of decentralized applications.

Even when processing is centralized, however, field-based computation models may prove useful. The same parallelism that allows them to be executed in a decentralized manner can also allow a field-based computation to be implicitly parallelized for faster processing on a server or cloud system. Finally, its adaptability to changes in resolution may also allow for mixed models, in which a fast, coarse model is computed in a decentralized manner to ensure rapid and non-disrupted services, while a higher-precision model is computed with somewhat more delay in the cloud, and supplied as it becomes available.

## 5 The View Forward

At a fundamental level, sensor network applications are generally well-represented as computation on space-filling fields of sensor information. The details of network interactions used to implement them are only important insofar as they support or interfere with this computation, and thus those details should be separated as much as possible from the specification of the sensor network application.

Using a field-based model of computing can facilitate this separation. When combined with libraries of self-organizing building block algorithms, this can greatly simplify the implementation of decentralized network applications, thereby expanding the applicability of sensor network applications in large-scale and difficult domains such as natural disaster response, mass events, and control of cyber-physical systems. Software support for making use of these methods is still early, but the simple mathematical core of field calculus and the open-world Java integration supported by Protelis significantly lower the cost of adoption compared to prior aggregate programming methods. With this better understanding of the formal foundation of sensor network applications and their consequences, we may thus look forward toward a future in which sensor network applications are increasingly scalable, resilient, and tightly integrated with the everyday operations of society.

## Acknowledgments

The views expressed in this manuscript have benefited from a number of valuable conversations with the participants of the Dagstuhl Seminar “Geosensor Networks: Bridging Algorithms and Applications,” organized in December 2013 by Matt Duckham, Stefan Dulman, Jorg-Rudiger Sack, and Monika Sester.

This work has been partially supported by the EU FP7 project “SAPERE - Self-aware Pervasive Service Ecosystems” under contract No. 256873 (Viroli), by the Italian PRIN 2010/2011 project “CINA: Compositionality, Interaction, Negotiation, Autonomicity” (Viroli), and by the United States Air Force and the Defense Advanced Research Projects Agency under Contract No. FA8750-10-C-0242 (Beal). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views, opinions, and/or findings contained in this article are those of the author(s)/presenter(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. Approved for public release; distribution is unlimited.
References


Section 2: Event Reports
Maps are rapidly becoming fully interactive interfaces to geographic information. We use them as tools to plan our journeys, to decide where we are going to live, to visualize data, or for gaming. We want to inspect relationships between entities, navigate through large result sets, and to quickly identify the best options amongst many alternatives. Whatever it is, we need to express our needs such that algorithms can provide the answer. However, working with maps is still surprisingly awkward. Simple queries can require a lot of interaction and workarounds, formulating complex queries is sometimes not even possible. To transform maps into intelligent and interactive interfaces, we need to anticipate users, contexts, and tasks and adapt interaction and visualization towards their needs and capabilities. MapInteract provides a platform to discuss these challenges and to shape the picture of maps of the future.

MapInteract 2014 (http://www.mapinteract.org) was held in conjunction with the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL 2014) on November 4, 2014 in Dallas, Texas, USA. During this highly interactive workshop we explored the future of maps as fully interactive interfaces. We were discussing contributions addressing different facets of interactivity with maps: from interaction design and user studies to algorithms enabling interaction with complex spatio-temporal data. With MapInteract 2014 we created space for exchanging ideas, discussion, and demonstrations between researchers and practitioners across disciplines and boundaries.

MapInteract 2014 has received 10 submissions of which 8 papers were accepted for presentations and/or demonstration. MapInteract 2014 was a one-day workshop consisting of three sessions: (1) Navigation and Routing, (2) Labeling and Demo, and (3) Interaction and Geo-Visualizations.

We would like to thank the authors for publishing and presenting their papers in MapInteract 2014, and the program committee members for their professional evaluation and help in the paper review process.
Big data is emerging as an important area of research for data researchers and scientists. This area has also seen significant interest from the industry and federal agencies alike, as evidenced by the recent White House initiative on “Big data research and development”. Within the realm of big data, spatial and spatio-temporal data is one of fastest growing types of data. With advances in remote sensors, sensor networks, and the proliferation of location sensing devices in daily life activities and common business practices, the generation of disparate, dynamic, and geographically distributed spatiotemporal data has exploded in recent years. In addition, significant progress in ground, air- and space-borne sensor technologies has led to an unprecedented access to earth science data for scientists from different disciplines, interested in studying the complementary nature of different parameters. Today, analyzing this data poses a massive challenge to researchers.

In 2012, we organized the first workshop on Analytics for Big Geospatial Data (BIGSPATIAL 2012) which was highly successful in bringing together researchers working in this area for a day long program consisting of several invited and technical talks. This was followed by an equally successful workshop held in 2013 (BIGSPATIAL 2013).

Building on the success of the previous editions to bring together researchers from academia, government and industrial research labs that are working in the area of spatial analytics with an eye towards massive data sizes, the 3rd workshop on Analytics for Big Geospatial Data (BIGSPATIAL 2014) was held in conjunction with the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL 2014) on November 4th, 2014. The main motivation for this workshop stems from the increasing need for a forum to exchange ideas and recent research results, and to facilitate collaboration and dialog between academia, government, and industrial stakeholders. We hope that this workshop provides a platform for researchers and practitioners engaged in addressing the big data aspect of spatial and spatio-temporal data analytics to present and discuss their ideas.

This year we received 13 technical submissions out of which 8 were selected for full presentations and 2 were selected for short presentations. The technical program also consisted of three keynote talks from well-known experts from academia and industry. The keynote speakers included Prof. Mohamed Mokbel from University of Minnesota, Dr. Siva Ravada from Oracle Spatial, and Dr. Erik Hoel from ESRI. The BIGSPATIAL workshop series will continue to provide a leading international forum for researchers, developers, and practitioners in the field of data analytics for big geospatial data to identify current and future areas of research.

We would like to thank the authors of all submitted papers. Their innovation and creativity has resulted in a strong technical program. We are highly indebted to the program committee members, whose reviewing efforts ensured in selecting a competitive and strong technical program. We would like to express our sincere gratitude to the invited speakers.
The ACM Special Interest Group on Spatial Information (SIGSPATIAL) addresses issues related to the acquisition, management, and processing of spatially-related information with a focus on algorithmic, geometric, and visual considerations. The scope includes, but is not limited to, geographic information systems (GIS).

The Association for Computing Machinery (ACM) is an educational and scientific computing society which works to advance computing as a science and a profession. Benefits include subscriptions to *Communications of the ACM*, MemberNet, TechNews and CareerNews, full and unlimited access to online courses and books, discounts on conferences and the option to subscribe to the ACM Digital Library.

- SIGSPATIAL (ACM Member) ................................................................. $ 15
- SIGSPATIAL (ACM Student Member & Non-ACM Student Member) ............................................. $ 6
- SIGSPATIAL (Non-ACM Member) .............................................................. $ 6
- ACM Professional Membership ($99) & SIGSPATIAL ($15) .................................................. $114
- ACM Professional Membership ($99) & SIGSPATIAL ($15) & ACM Digital Library ($99) ............... $213
- ACM Student Membership ($19) & SIGSPATIAL ($6) .............................................................. $ 25

**Payment Information**

Name ____________________________

ACM Member # ____________________

Mailing Address ______________________

City/State/Province ____________________

ZIP/Postal Code/Country ________________

Email ________________________________

Mobile Phone __________________________

Fax _________________________________

Mailing List Restriction
ACM occasionally makes its mailing list available to computer-related organizations, educational institutions and sister societies. All email addresses remain strictly confidential. Check one of the following if you wish to restrict the use of your name:

- ACM announcements only
- ACM and other sister society announcements
- ACM subscription and renewal notices only

Credit Card Type:  □ AMEX  □ VISA  □ MC

Credit Card #: __________________________

Exp. Date ___________________________

Signature ___________________________

Make check or money order payable to ACM, Inc

ACM accepts U.S. dollars or equivalent in foreign currency. Prices include surface delivery charge. Expedited Air Service, which is a partial air freight delivery service, is available outside North America. Contact ACM for more information.

**Questions? Contact:**
ACM Headquarters
2 Penn Plaza, Suite 701
New York, NY 10121-0701
voice: 212-626-0500
fax: 212-944-1318
email: acmhelp@acm.org

Remit to:
ACM
General Post Office
P.O. Box 30777
New York, NY 10087-0777

www.acm.org/joinsigs

Advancing Computing as a Science & Profession
The SIGSPATIAL Special

ACM SIGSPATIAL
http://www.sigspatial.org