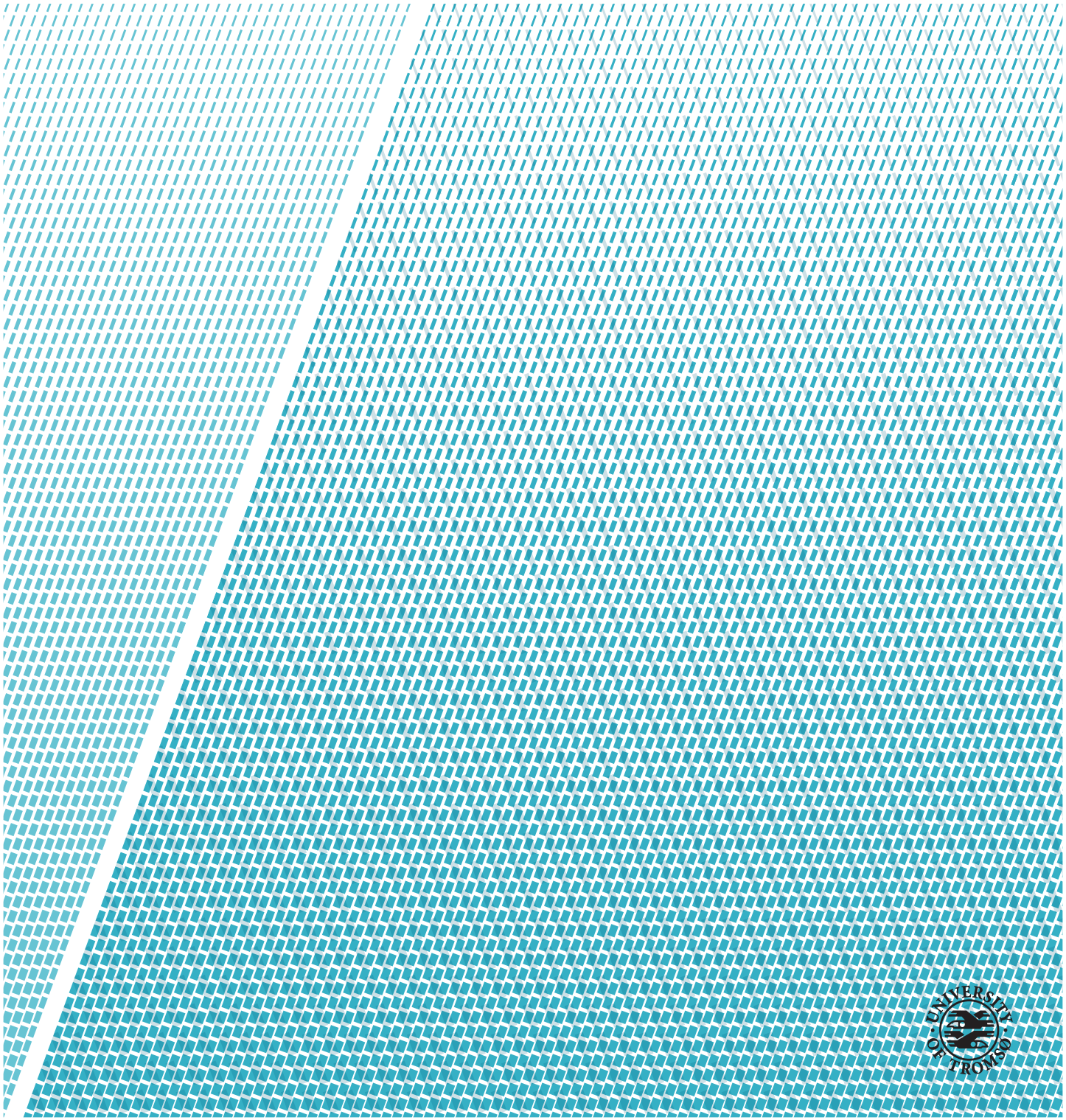
UiT
THE ARCTIC
UNIVERSITY
OF NORWAY

Faculty of Science and Technology
Department of Computer Science

From Physical to Virtual Sensors (PVS)

Camilla Stormoen

INF-3983 Capstone Project in Computer Science . . . December 2017



Abstract

W3 Whats wrong with the word? / motivation 1-3 setninger

Architecture - 1-3 setninger

Design- 1-3 setninger

Implementation - 1-3 setninger

Experiments - 1-3 setninger

Results - 1-3 setninger

Lessons learned/main conclusion - 1-3 setninger

Kutt heller etterpaa

This dissertation present/describe ...

Contents

Abstract **i**

List of Figures **v**

1 Introduction **1**
1.1 Motivation 1
1.2 Contributions 2
1.3 Assumptions 2
1.4 Limitations 2

2 (Background and) Related Work **3**
2.1 Virtual Sensors 4
2.2 Something 4

3 Architecture **5**
3.1 Physical Sensors and Data Storage 6
3.2 Fused Data 6
3.3 Virtual Sensors 6
3.4 Result from Virtual Sensor to User 6

4 Design **7**
4.1 Data Storage and Fused Data from Sensors 7
4.2 Fused Data from Sensors 8
4.3 Virtual Sensors 9
4.4 User 9

5 Implementation **11**

6 Evaluation **13**
6.1 Experimental Setup 14
6.2 Something!? 14
6.3 Results 14

7 Discussion **15**

7.1 Architecture 15
7.1.1 Other solutions 15
7.2 Design 15
7.2.1 Other solutions 15

8 Contributions **17**

9 Conclusion **19**
9.1 Contributions? 19
9.2 Future Work 19

10 Future Work? **21**

11 Appendix? **23**

Bibliography **25**

List of Figures

3.1 Figure showing the system architecture. 5

4.1 Figure showing the system design. 8

1 Introduction

- Mention focus on camera-sensors/data, and not other sensors?!
- Talk a little bit about COAT in general?

This project will develop an abstraction for virtual sensors, and do a prototype of the abstraction on a set of computers with physical sensors.

The purpose is to provide for a more powerful and flexible sensor in the COAT monitoring of the arctic tundra. As such, a fox feeding station is the usage domain to be used for the prototype.

1.1 Motivation

The motivation!

- W3
- Problem definition: This project investigated ... x, with the purpose of y.

The motivation behind this project is that no single sensor may cover the sensing needs, and that sensing needs can change rapidly over time. Consequently, there is a need for sensor fusion, and allow for combining sensors at different

computers.

1.2 Contributions

What was the contribution?

1.3 Assumptions

AVGRENSE, VIKTIG!! Something about motivation and stuff

1.4 Limitations

AVGRENSE, VIKTIG!!

- Mention focus on camera-sensors/data, and not other sensors?!

/2

(Background and) Related Work

- Taking Sensor Networks from the Lab to the Jungle
- Wireless Sensor Networks for Habitat Monitoring
- Building Virtual Sensors and Actuators over Logical Neighborhoods
- A virtual sensor system for user-generated, real-time environmental data products
- Integrating modeling and smart sensors for environmental and human health
- Capability representation model for heterogeneous remote sensing sensors: Case study on soil moisture monitoring
- Dice: Monitoring Global Invariants with Wireless Sensor Networks
  Ştefan Gună, Luca Mottola, and Gian Pietro Picco. 2014. DICE: Monitoring global invariants with wireless sensor networks. ACM Trans. Sensor Netw. 10, 4, Article 54 (April 2014), 34 pages. DOI: http://dx.doi.org/10.1145/2509434

2.1 Virtual Sensors

A virtual sensor is a constructed sensor in contrary to a physical sensor as described by S. Kabadayi et. al.[1]. They are used in place of the real sensors where they read real physical sensor data and calculate the outputs by using some processing models.

Recent research efforts have focused on simple in-network data aggregation techniques. Project targeted directly for sensor networks have often explored representing the sensor network as a database. Two demonstrative examples are TinyDB [6] and Cougar [7]. Generally these approaches enable applcations with data requests that follow out from a central point (e.g. a base station) and create routing trees to funnel replies back to this root. These approaches focus on performing intelligent in-network aggregation and routing to reduce the overall energy cost while still keeping the semantic value of data high. in both approaches, data aggregation is specified using an SQL-like language. Queries cannot be used to merge different data types; only homogeneous data aggregation is possible. In contrast, the virtual sensor approach offers simple programming interface, supports multiple access points, and offers raw and heterogeneous in-network data processing.

In this project the biologists want to search for pictures of a specific animal from different locations in the Arctic Tundra. A virtual sensor would collect data from the physical sensors and then return those images that are equivalent to the biologists input.

Ciciriello et. al.[2] describes virtual nodes which are programming abstractions that should simplify the development of decentralized wireless sensor network (WSN) applications. Their system enable access to the data by a given set of sensors and making it into one virtual sensor that a programmer can interact with. The physical nodes are abstracted and specified using logical neighborhoods [4][5]. The nodes are combined in a logical neighborhood that are specified based on their characteristics by the programmer.

TinyOS [3]

Our work relies on physical sensor data which is already in a data storage, and not directly from the physical sensors.

2.2 Something

3

Architecture

This chapter describes the architecture of the system. There are 6 components in the system: physical sensors, data storage, fused data, virtual sensors and the user. However, the main components in the system are the data storage, the fused data, the virtual sensors and the user. The architecture of the system is presented in Figure 3.1.

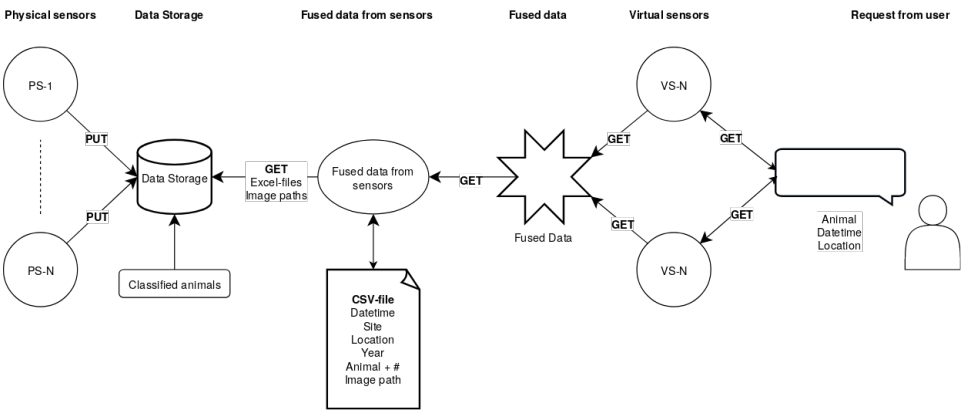


Figure 3.1: Figure showing the system architecture.

3.1 Physical Sensors and Data Storage

The physical sensors transmit their data to the data storage, so the data storage consists of several set of images from the different bait-camera sensors. These pictures are organized by the year they where taken and also where the camera was placed in the Arctic Tundra in Finnmark, Norway. The data store also contains several excel-sheets containing information about each picture structured by the location of the physical bait-camera sensor position.

3.2 Fused Data

The fused data retrieves it's data from the data storage and store the fused data into a Comma-Separated Value (CSV) file on demand. The fused data is the image-path from all the directories in the data storage combined with necessary information from the excel-sheet such as date-time, location, site, year, what kind of animal was in the image and also how many animals there was.

3.3 Virtual Sensors

The virtual sensors are divided multiple sensors representing different animals that scientists are interested in, e.g. one raven-sensor, one red fox-sensor and one golden eagle-sensor. The user types in what animal it wants to see, where it is and the date-time and the search is redirected to the sensor related to that specific animal. The virtual sensor receive its result from the fused data from the CSV-file.

3.4 Result from Virtual Sensor to User

A user wants to retrieve images from the data store. The user interacts with a user application. This application takes care of the communication to the virtual sensors. When a user gives a command, the user application sends it to one of the virtual sensors, the virtual sensor retrieve data from the fused data as described in the section above, and deliver the response back to the user. The image(s) are displayed through an image-vision.

4

Design

Client/Server, p2p, put/get, pub/sub, protokoller etc.. BESKRIV INTERAKSJONEN MELLOM ENHETENE!!

Virtual sensors probably uavhengige prosesser, ikke threads ettersom man evt vil addere flere sensorer og unnga a starte alle sensorer på nytt igjen.. Er de virtuelle sensorene servere eller client/publisher?

In this chapter we will look at the design of the environment. We will present the design of the data storage, the fused data the virtual sensors and the user application. Figure 4.1 shows the design of the system. The arrows indicates the communication lines. The dotted arrows show how the data storage is structured with files and pictures.

4.1 Data Storage and Fused Data from Sensors

The dataset from the physical sensors contains over 1.6 millions of pictures from 2011 to 2016 taken from the Arctic Tundra in Finnmark, Norway. The physical sensors are placed at different areas in the Arctic Tundra such as Nordkynn, Komag, Nyborg, Stjernevann and Gaissene. The excel-sheets contains from 17 000 to 70 000 rows with information about each picture taken.

The excel-sheet in the data storage contains a pictures location, date-time,

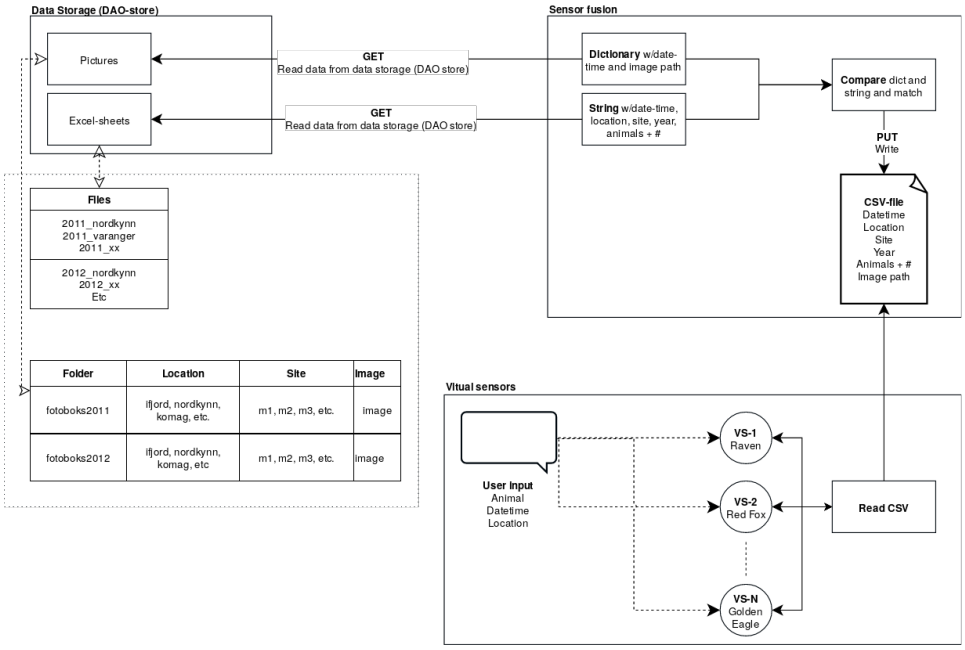


Figure 4.1: Figure showing the system design.

site and year, but it did not contain any filename or a filepath. We then had to find a method to find out which images correspond to which field/row in the excel-sheet. Fortunately, each image had metadata stored in Exchangeable Image File Format (EXIF). We recursively traversing the directories where pictures are located and read date and time for each picture which is stored in a Python dictionary, like a key-value store, with date and time as key and image-path as value.

Date and time from each image is extracted and stored in a dictionary together with the necessary information from the excel-sheet.

4.2 Fused Data from Sensors

Rekursiv traversering av directories og leser metadata fra bildene (ca 1,6 mill bilder)) - Lagres i en dictionary hvor datetime og sted er key og image pathen er value).

FRA HÅVARDS MASTER: The csv files did not contain any filenames, only image metadata and animal classifications, so we had to get creative to find out which images the annotations corresponded to. Fortunately, each image had metadata stored in Exchangeable Image File Format (exif). We created

a Python script to extract the date and time from each image, along with camera information, to match them with the annotations. This was quite a time-consuming task because of the number of images to process.

Disse sammenlignes og de som matcher blir skrevet til en CSV-fil.

4.3 Virtual Sensors

4.4 User

The user-application is the connection between the user and the virtual sensors. It gets input from the user and sends a request to the virtual sensor based on its input. The user-applications goal is to make an interface that is easy to use and to show images that the user want to see. The user only need to specify the following 3 paramteres for the virtual sensor:

Animal

Date-time

Location

- Animal:
- Date-time:
- Location:

5

Implementation

Threads, data structures, language ... Pandas (dataframe [1] [2]), CV2 (show image), exifread, Python 2.7, missing testing (CPU, memory, time?)

The system is implemented and written in Python 2.7[3] because .. (frameworks available in this language??).

To visualize/show pictures, a Python library called OpenCV [4] was implemented. To read exif/metadata from pictures, we used a Python library called exifread 2.1.2 [5].

1. https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.html
2. http://pandas.pydata.org/pandas-docs/stable/
3. https://www.python.org/
4. https://opencv-python-tutroals.readthedocs.io/en/latest/
5. https://pypi.python.org/pypi/ExifRead

6

Evaluation

metrics, define (CPU, memoury, lantecy.), benchmarks (mirko, kernel... How to measure, where done, PSEUDOCODE

- Time: Finding folders and metadata takes: 1:43:13.488799, Reading excel file takes: 0:00:17.413845, Comparing takes: 4:43:30.705587, Overall time is 6:27:01.608355. Med alle bilder m/metadata og hele fotoboks2011_nordkynn_nordkynn.2011.xlsx.

- New time: Finding folders and metadata takes: 1:46:17.406581 Reading excel file takes: 0:01:07.686779 Comparing takes: 19:03:11.177869 Overall time is 20:50:36.271415 Med alle bilder m/mETADATA og hele nordkynn og varanger

- "Concurrent" python find_folder_ok_concurrent.py Finding folders and metadata takes: 1:30:50.250732 Reading excel file takes: 0:00:18.127597 Comparing takes: 4:18:33.368199 Overall time is 5:49:41.746759 – Pool(processes=4) FRA fotoboks2011_nordkynn_nordkynn.2011.xlsx

- "Concurrent" igjen Finding folders and metadata takes: 1:44:18.788013 Reading excel file takes: 0:00:17.269817 Comparing takes: 3:54:23.483186 Overall time is 5:38:59.541248 – Pool(processes=100) FRA fotoboks2011_nordkynn_nordkynn.2011.xlsx

  This chapter describes the experimental setup and metrics used to evalu-

ate the implemented system.

6.1 Experimental Setup

All experiements was done on a Lenovo ThinkCenter with an Intel® Core™ i5-6400T CPU @ 2.20GHz × 4, Intel® HD Graphics 530 (Skylake GT2), 15,6 GiB memory and 503 GB disk. It ran on Ubuntu 17.04 64-bit.

6.2 Something!?

6.3 Results

What does the result say? Each experiment, result, meaning

/7 Discussion

Idea, architecture, design, results, other solutions, "arch have scale-problem?".

7.1 Architecture

7.1.1 Other solutions

- Virtual sensors probably uavhengige prosesser, ikke threads ettersom man evt vil addere flere sensorer og unnga a starte alle sensorer på nytt igjen..
- Updates of "database" in background, not on demand -> future work

7.2 Design

7.2.1 Other solutions

- hash-map instead of multiple lists with different animals. hash on animal and/or place
- parallel/concurrent program when doing metadata-collecting and read-

ing from excel-sheet and writing to csv

- FRA HÅVARDS MASTER: The csv files did not contain any filenames, only image metadata and animal classifications, so we had to get creative to find out which images the annotations corresponded to. Fortunately, each image had metadata stored in Exchangeable Image File Format (exif). We created a Python script to extract the date and time from each image, along with camera information, to match them with the annotations. This was quite a time-consuming task because of the number of images to process.

8

Contributions

Combine with conclusion??

9

Conclusion

9.1 Contributions?

9.2 Future Work

10

Future Work?

11

Appendix?

readme, source code, dataset measurement RAW

Bibliography

[1] S. Kabadayi and A. Pridgen and C. Julien *Virtual Sensors: Abstracting Data from Physical Sensors*, 2006, in *2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks(WoWMoM'06), 6 pp.-592*.

[2] Ciciriello, Pietro and Mottola, Luca and Picco, Gian Pietro *Building Virtual Sensors and Actuators over Logical Neighborhoods*, 2006, in ACM, 19–24. *http://doi.acm.org/10.1145/1176866.1176870*.

[3] Hill, Jason and Szewczyk, Robert and Woo, Alec and Hollar, Seth and Culler, David and Pister, Kristofer *System Architecture Directions for Networked Sensors*, 2000, in *ASPLOS-IX: Proc. of the 9 nt Int. Conf. on Architectural Support for Programming Languages and Operating Systems*, pages 93–104.

[4] Mottola, Luca and Picco, Gian Pietro *Logical Neighborhoods: A Programming Abstraction for Wireless Sensor Networks*, 2006, in *Distributed Computing in Sensor Systems: Second IEEE International Conference, DCOSS 2006, San Francisco, CA, USA, June 18-20, 2006. Proceedings*, pages 150–168. *http://.............*

[5] Mottola, Luca and Picco, Gian Pietro *Programming Wireless Sensor Networks with Logical Neighborhoods*, 2006, in *Proceedings of the First International Conference on Integrated Internet Ad Hoc and Sensor Networks*, series = InterSense '06, pages 150–168. *http://doi.acm.org/10.1145/1142680.1142691*.

[6] Madden, Samuel R. and Franklin, Michael J. and Hellerstein, Joseph M. and Hong, Wei *TinyDB: An Acquisitional Query Processing System for Sensor Networks*, March 2005, in *ACM Trans. Database Syst. Vol. 30, Nr. 1*, pages 122–173. *http://doi.acm.org/10.1145/1061318.1061322*.

[7] Yao, Yong and Gehrke, Johannes *The Cougar Approach to In-network Query Processing in Sensor Networks*, September 2002, in *SIGMOD Rec., Vol. 31, Nr. 3*, pages 9–18. *http://doi.acm.org/10.1145/601858.601861*.