Bio-Inspired Field Estimation with Wireless Sensor Networks

Daniel de O. Cunha, Rafael P. Laufer, Igor M. Moraes, Marco D. D. Bicudo,
Pedro B. Velloso, and Otto Carlos M. B. Duarte
Grupo de Teleinformática e Automação - PEE/COPPE - DEL/POLI
Universidade Federal do Rio de Janeiro - Rio de Janeiro, Brazil

***Abstract*— This paper proposes and analyzes a bio-inspired scheme to field estimation using wireless sensor networks. The proposed scheme exploits the temporal pattern of the sensed process to reduce the number of samples sent back by a sensor node to the sink and decrease the energy consumption in data transmission. The proposed scheme is orthogonal to the techniques that reduce the spatial density of collected samples deactivating nodes with similar measurements. Thus, the proposed scheme can be used along with these techniques. Results show that for very regular processes the scheme can reduce around 90% the total amount of samples sent in the network and even for less regular processes the proposed scheme can reduce between 10 and 20% the total amount of samples sent with small reconstruction errors.**

I. Introduction

Field estimation is an important application of wireless sensor networks. This type of application deploys sensor nodes in a specific region to remotely sense space-temporally variable processes. The spatial and temporal frequencies of sampling directly impact the quality of the estimation. The spatial frequency depends on the number of nodes and how these nodes are distributed, while the temporal frequency depends on the rate of sampling of the active nodes. High frequencies consumes more resources, but low frequencies may reduce the accuracy of the process estimation [1]. Therefore, there is a tradeoff between the frequency and the number of samples transmitted, which is related to the energy consumption of the network.

The most common solution to reduce the number of samples transmitted to the sink is to identify areas where different nodes present similar readings and reduce the sampling spatial frequency by deactivating some of these nodes [2], [3], [4]. The idea is to expand the network lifetime collecting fewer samples in regions with smooth spatial variations. Thus, the transmission of redundant information is reduced and the high cost of data transmission is avoided [5]. Nevertheless, this approach is not efficient in regions of the field with sharp spatial variations, or borders. These regions usually represent important aspects of the process [6] and need to have many nodes actives. There is, however, another sampling dimension to regard: the temporal dimension. Little effort has been made to know when active nodes must collect and transmit samples. A feature that can be exploited is the temporally regular behavior of many physical processes. Temperature, for instance, generally has a similar behavior in consecutive days.

We propose to allow nodes to identify patterns in the behavior of the sensed processes and report only uncommon measures. This environment-aware behavior is similar to the response of living beings to the surrounding events. People and animals are continually receiving *stimuli*; however, it is impossible to handle consciously all these *stimuli*. The organisms develop the notion of periphery and center of attention [7]. While the periphery is handled in a sub-conscious manner, the center of attention is the event consciously treated. Generally, an event migrates from the periphery to the center of attention when it differs much from the periphery as a whole.

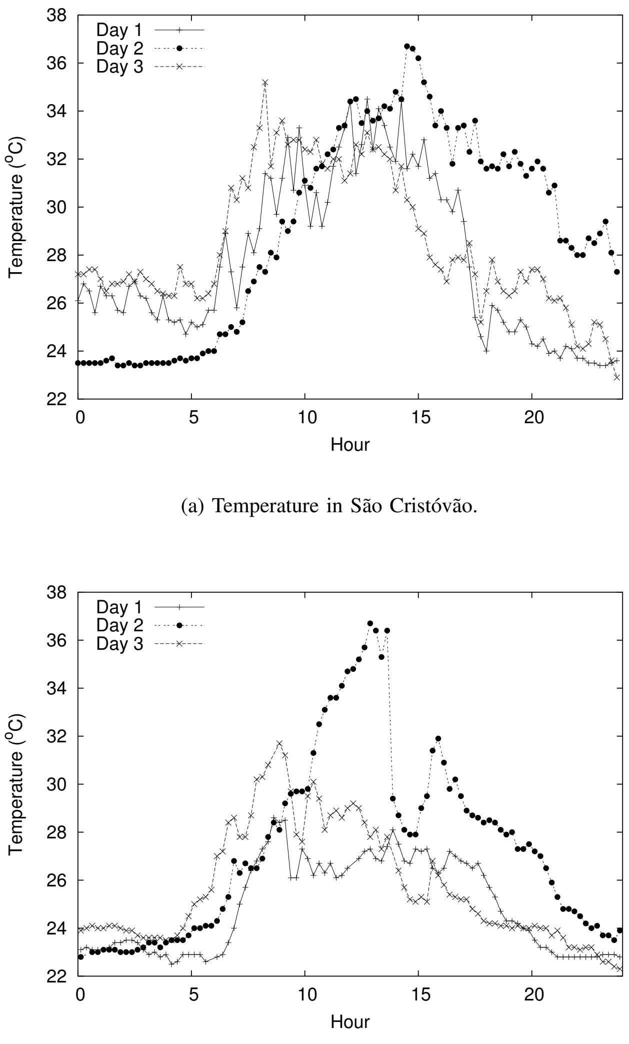
This paper proposes and analyzes a bio-inspired scheme to exploit specific features of the monitored processes in order to reduce the number of transmitted samples. The proposed scheme is fully distributed as each node identifies its own periphery (Section II). Thus, each node sends to the sink only the samples that differ from the usually sensed by the node. This scheme can save resources for nodes located in border regions as in smooth varying regions. This temporal technique is orthogonal to space-frequency-reduction techniques and both techniques can be used together to improve the system performance. This paper is organized as follows. Section II defines more clearly how some features of the physical processes can be exploited and details the proposed scheme. The results of the simulations of the scheme are showed in Section III. Finally, Section IV concludes this paper.

II. Proposed Scheme

The proposed scheme is based on the construction of a periphery of attention by each sensor node. It is necessary that the monitored process present an expected time behavior. Indeed, many physical processes present regular behaviors, sometimes with well-defined cycles. Fig. 1 shows the temperature measured in two different places of Rio de Janeiro during three consecutive days [8].

It is clear from Fig. 1 a regular behavior where the temperature is low in the morning and rises near noon. The temperature falls in the afternoon and reaches low values again at night. Relating to the idea of periphery, a temperature curve that is similar to the average of these curves of Fig. 1 could remain in the periphery. Otherwise, a curve differing above a certain threshold must be explicitly taken into account. While

This work has been supported by CNPq, CAPES, FAPERJ, RNP, FINEP e FUNTTEL.



(a) Temperature in São Cristóvão.

(b) Temperature in Guaratiba

Fig. 1. Temperatures in two distinct places of Rio de Janeiro.

the three curves in Fig. 1(a) present very similar behavior, the Day 2 curve of Fig. 1(b) has significant differences from the Day 1 and Day 3 curves. The proposed scheme aims at conserving resources not sending information when situations like the one in Fig. 1(a) occurs. On the other hand, the accuracy of the estimation is preserved sending information for situations like the Day 2 curve in Fig. 1(b). Therefore, the sink assumes the behavior is the one expected if no update sample is received. Furthermore, update samples are included in the process reconstruction, keeping low the estimation error in situations like Day 2 curve of Fig. 1(b). These samples sent due to their difference with the regular behavior are called *refining samples*.

The periphery construction and refining sample generation can take different forms. The first stage is to determine the periodicity of the regular behavior. One option to do this is to collect samples for a relatively long time and calculate the autocorrelation of the vector of samples. Based on the autocorrelation, the period of the process can be obtained. Once the period of the regular behavior is identified, the node can begin defining its own periphery, or expected behavior, as follows:

$$PB_i = P_{i-1} \times \alpha + PB_{i-1} \times (1 - \alpha), \quad (1)$$

where PB_j is the vector containing the expected behavior during period j, P_{i-1} is the vector with the samples collected during period $(i-1)$, and α is the weight of the last period obtained in the expected behavior. Thus, as α increases, the higher will be the importance of more recent periods on the expected behavior and smaller will be the influence of the historic behavior of the process.

In this paper we focus temperature monitoring, which clearly has a daily periodicity. Temperature presents also an annual periodicity, but our analysis is based on the daily periodicity. Hence, the sensor nodes must identify a daily-expected behavior, updated every day. The decision over refining samples is done based on this behavior. Also, it is necessary to provide the sink with enough information to correctly reconstruct the field. In order to do that, the sink needs to know an expected behavior, which is assumed to occur when no refining samples are received. The node must send periodically an updated expected behavior to the sink. Thus, until a new expected behavior is received, the sink assumes the process behaves like the last expected behavior received added to eventual refining samples. These refining samples will replace the samples of the expected behavior for the hour informed by the sensor as it sends the refining sample. The sensor node must decide on sending or not refining samples based on the last expected behavior vector sent to the sink. This procedure maintains the consistency between the measured and the reconstructed information. Therefore, the sensor verifies if the measured value differs above certain threshold from the sample for the specific hour in the last expected behavior sent. If this difference is higher than the configured threshold, the sensor sends the refining sample and the timing information to the sink.

It is worth to note that each sensor node builds and analyzes its periphery on its own time basis, which dispenses synchronization between sensor nodes. The timing information for the reconstruction at the sink is relative to the time basis of the specific sensor node. Moreover, if each day the sensor sends the expected behavior for the next day, the system will achieve no gain. Therefore, the sensor periodically sends expected behavior updates periodically but at a frequency low enough to achieve some gain. Fig. 2 shows the daily procedure, where DB_j is the vector with the expected behavior during day j, D_i the vector with the measurements of day i, $last\_update$ is the vector with the last expected behavior sent to the sink, and $X(k)$ is the k-th element of vector X.

The expected behavior calculus of Fig. 2 is slightly different from the one showed in Eq. 1 in order to send, in the update day, the expected behavior for that day ($last\_update = DB_i$) instead of the measures for that day (D_i). This is only a design option. The scheme could send the expected behavior and the refining samples for the update day. As α get higher and the process variation gets lower, these two approaches become

```
DB i = α D i + (1−α) DB i−1
If update time
    last_update = DB i
    Send last_update
Else
    For all k samples in D i do
        If |D i(k) − last_update(k)| > ||last_update(k)| * configured_error
            Send D i(k)
```

Fig. 2. Daily procedure for each sensor node.

equals. Next section details the simulations of the proposed scheme.

III. Simulations

The efficacy of the proposed scheme is evaluated by simulation considering the reduction in the total number of samples sent to the sink. The simulations presented here are based on the temperature monitoring problem described in Section II. During the simulations, three main variables are considered: the update frequency, the α parameter, and the maximum error without refining.

An important part of the analysis of the proposed scheme is the data generation. The data model must represent the main features of the application with respect to the results of the scheme [9]. Section III-A give some insights of the data generation procedure used to analyze the proposed scheme.

A. Data Modeling

The data is generated based on observation of real-site-collected data like the ones showed in Fig. 1. Observing these collected data, it is possible to note important features in the temperature evolution during one day and during consecutive days. There are two distinct features in the temperature evolution during one day: the higher temperature near noon and the sharper variation of the temperature in these moments of higher average temperature. The temperature in consecutive days presents some relation in its average and in the difference between the lower and the higher temperature of the day.

It is worth noting that the goal of this data model is not to create real temperature curves, but to create curves that exhibit the same main behavior of temperature curves. Generating data this way makes it possible to better evaluate the proposed scheme to the target application.

First of all, the basis for the construction of the daily behavior is the function $-cos(x)$. This function exhibits its higher values near the middle of the period. A bias is added to this function to make the average temperature behave similar to Fig. 1(a). In order to model the behavior of consecutive days, this bias added at day i ($bias(i)$) is defined as the bias of the anterior day $i-1$ ($bias(i-1)$) plus a Gaussian random variable with 0 mean and standard deviation σ_1. Greater relations between the average temperature of consecutive days are achieved with smaller values of σ_1. The same procedure is used to model the amplitude of the function $-cos(x)$ for day i ($amp(i)$) based on the amplitude of day $i-1$ ($amp(i-1)$). This relation between the two amplitudes represents the relation between the difference of the higher and the lower temperature of the consecutive days.

Finally, another Gaussian random variable with 0 mean and standard deviation σ_2 is added to each point of the function $-cos(x)$ representing the temperature in different hours of the day. In order to reproduce the sharper variation in hours with higher average temperatures, this random variable is multiplied by a constant C proportional to the value of the $-cos(x)$ function at the specific hour. All the random variables have their maximum and minimum values limited. This avoids unreal differences of temperature between consecutive days.

The daily data has 96 samples, which means data collection at 15 minutes intervals. The values of σ_1 and σ_2 are varied during the simulation to evaluate the effect of these parameters over the proposed scheme. The next section presents further details of the simulations and discusses the achieved results.

B. Results

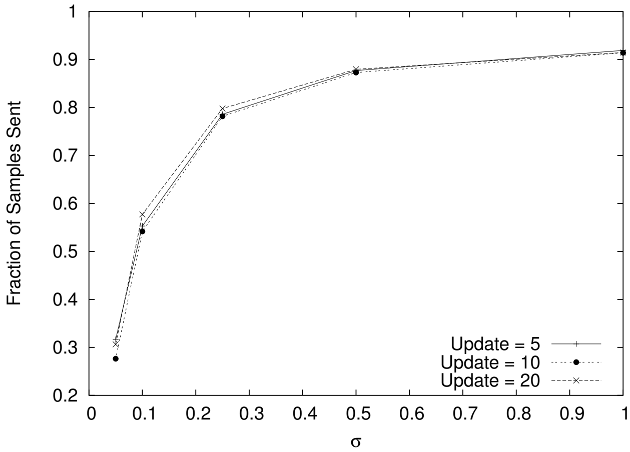
The simulations are based on data generated as described in Section III-A representing the readings of a sensor. The proposed scheme is applied to this data set and the fraction of the total samples that must be actually sent to the sink is obtained. The smaller this fraction is, the better the proposed scheme efficacy. The simulation starts with the daily (96 samples) periodicity defined. Thus, the analysis evaluates the steady state operation of the scheme. Moreover, the identification of the regular period is independent of the parameters used in the scheme and depends only of the data set used.

There are three main parameters in the proposed scheme: the update frequency, the α factor, and the tolerated sample error. Therefore, these parameters are varied during the simulation in order to better understand their effects. The α factor is bounded to 1. In all simulations, the update frequency is one expected behavior vector sent at each $Update$ days. Thus, higher values of $Update$ means lowers update frequencies. The tolerated sample error is equal to the parameter $maximum\_error$ times the expected behavior of the specific hour. All the results shown have a confidence interval smaller than 3% of the average value for a confidence level of 99%.

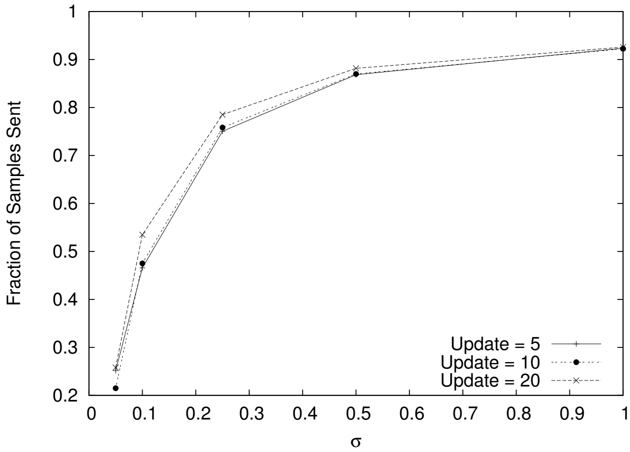
The first analysis concerns the impact of the uncertainty of the sensed process over the fraction of samples sent. This is done evaluating the effects of the variation of σ_1 and σ_2 over the number of samples sent. Fig. 3 shows the fraction of the samples sent when σ_1 and σ_2 varies together (σ) for a tolerated error of 1%.

Analyzing Fig. 3 it is possible to observe that the small value of the tolerated error severely reduces the influence of the α and $Update$ parameters. For processes with smaller variation (low σ) the fraction of samples sent tends to be $\frac{1}{Update}$. It is worth noting that the smaller update frequency used, $Update = 20$, sends less samples when the process does not vary much, but clearly has a worse performance when the variation of the process increases (higher σ). For this tolerated error, the fraction of sent samples approaches 90% rapidly with the increase of σ. Fig. 4 shows the results for a tolerated error of 5%.

As we can see from Fig. 4, with the increase of the tolerated error the parameters α and $Update$ have larger influence in the results. Thus, the variation of these parameters produces more

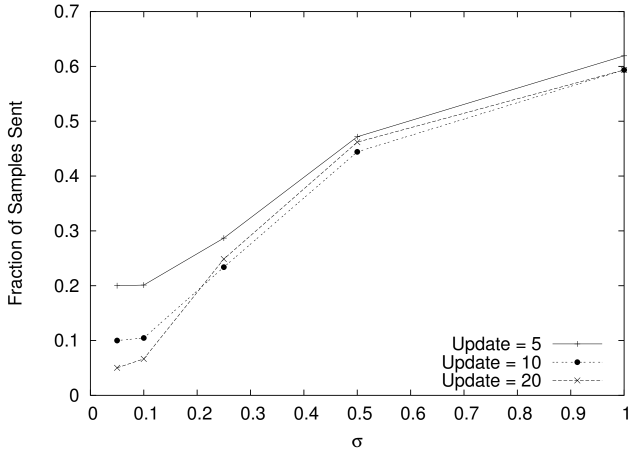


(a) $\alpha = 0.1$.

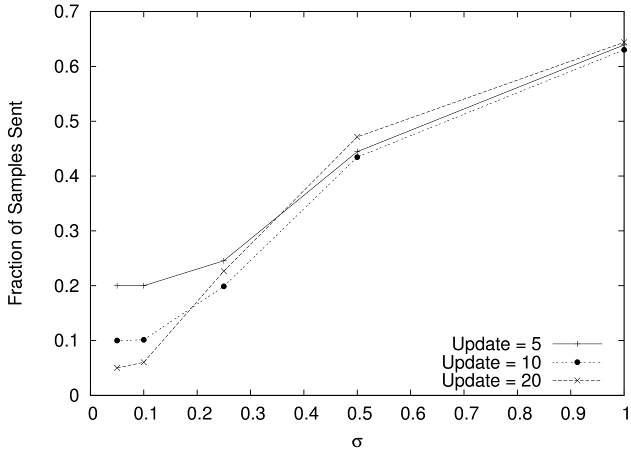


(b) $\alpha = 1$

Fig. 3. Samples sent varying σ for $maximum\_error$ = 0.01.



(a) $\alpha = 0.1$.



(b) $\alpha = 1$

Fig. 4. Samples sent varying σ for $maximum\_error$ = 0.05.

significant changes in the results. For a tolerated error of 5%, the proposed scheme reduces between 35 and 40% the number of samples sent for σ = 1. Moreover, for higher values of σ, the intermediate $Update$ value, 10, presents better results. This suggests that this intermediate value of $Update$ achieves a better tradeoff between the fixed part of the total samples sent ($\frac{1}{Update}$) and the samples sent due to the difference from the expected behavior.

The importance of the tolerated error makes it necessary to further investigate its influence over the results. Fig. 5 shows the results as a function of the tolerated error for $Update$ = 10 for an intermediate value for α.

Fig. 5 shows that as the tolerated error decreases, the variations of σ (σ_1 and σ_2 together) have worse impact on the scheme performance. For high tolerated errors, small variations of σ are negligible. It is worth to note the limit $\frac{1}{Update}$ to which the fraction of samples tends when the tolerated error grows or σ is reduced (more regular processes). Moreover, even for less regular processes, σ = 1, it is possible to reduce between 10 and 20% the total number of samples

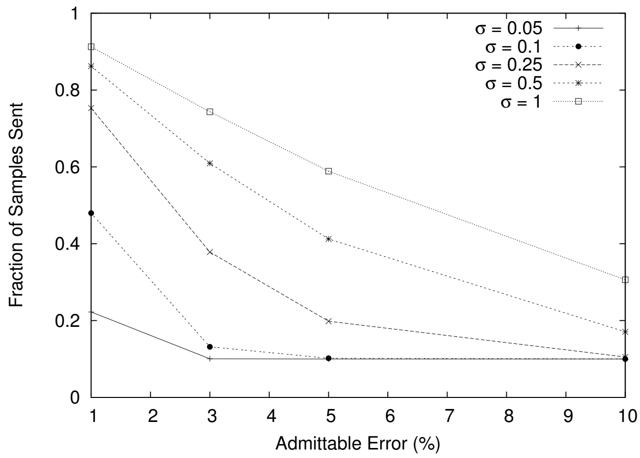
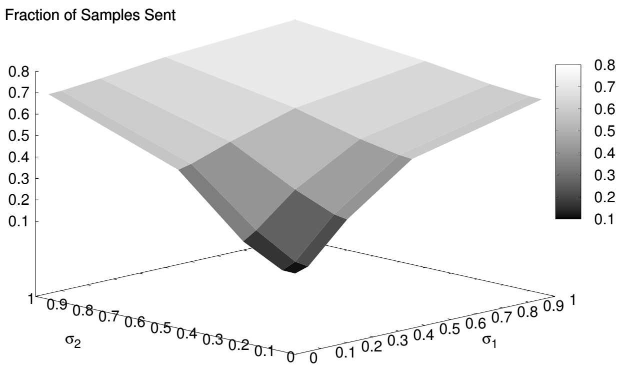


Fig. 5. Samples sent as a function of the tolerated error for $\alpha = 0.25$.

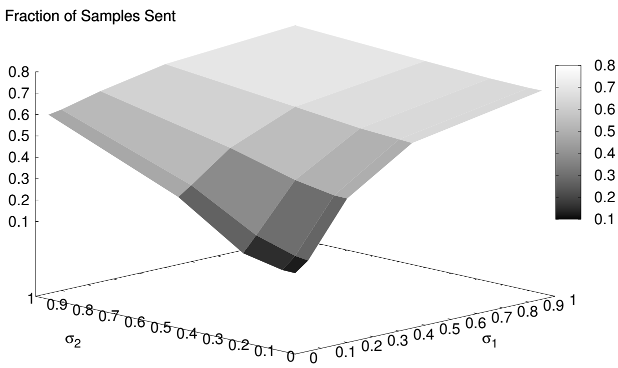
sent for tolerated errors as low as 1 and 3%.

Finally, we analyze the effects of σ_1 and σ_2 separately.

Fig. 6 is generated using an intermediate tolerated error value, 3%. This tolerated error keeps the sensitivity of the performance to the variations of σ in a reasonable level, avoiding the extreme sensitivity showed in Fig. 5 for the 1% case and the relative insensitivity of the 10% case.



(a) $\alpha = 1$.



(b) $\alpha = 0.1$

Fig. 6. Samples sent as a function of σ_1 and σ_2 for $maximum\_error$ = 0.03 and $Update$ = 10.

While the scheme for $\alpha = 1$ (Figura 6(a)) is almost equally sensitive to the variations of σ_1 and σ_2, when α is set to 0.1 (Figura 6(b)) there is a higher sensitivity to σ_1. This behavior is especially noticeable in the region of the graphs where σ_1 and σ_2 are low. The sensitivity of the scheme to σ_2 is similar because this parameter does not influences the relation between consecutive days. On the other hand, σ_1 dictates the relation between the behaviors of consecutive days and the α parameter aim at enhancing the performance of the scheme in different consecutive days relations.

IV. Conclusions

This paper proposes and analyzes a bio-inspired scheme to reduce the number of samples sent back by sensor nodes to the sink. The proposed scheme exploits specific features of the sensed physical process, identifying a regular behavior of the process, which will define the periphery of attention of the sensor node. Peripheral events, or events similar to the periphery, are no reported to the sink. This procedure reduces the data traffic and, consequently, the energy consumption in the network. The proposed scheme has the advantage of being suitable even for nodes at borders of the fields, which are unable to benefit from conventional techniques that reduce the spatial density of collected samples. Moreover, the proposed scheme can be used together with these conventional techniques to obtain greater gains.

The viability of the proposed scheme was analyzed based on a temperature monitoring application. The data generation was designed to reproduce the main features of this physical process with respect to the performance of the scheme. Results show a tradeoff between the number of times the node updates the expected behavior and the amount of samples sent due to high difference with the expected behavior. Severe restrictions on the tolerated error rapidly reduce the gain of the scheme as the process becomes less regular. Results show that the scheme must be tuned based on the regularity of the process to enhance the performance. For very regular processes the scheme can reduce up to 90% the total amount of samples sent in the network. For less regular processes the proposed scheme can still reduce between 10 and 20% the total amount of samples sent with small reconstruction errors.

As future works we intend to analyze the effects of packet losses in the network on the information reconstruction error and to evaluate the scheme based on data collected in real sites.

References

[1] A. Kumar, P. Ishwar, and K. Ramchandran, "On distributed sampling of smooth non-bandlimited fields," in *Information Processing In Sensor Networks - IPSN'04*, apr 2004, pp. 89–98.

[2] R. Willett, A. Martin, and R. Nowak, "Backcasting: adaptive sampling for sensor networks," in *Information Processing In Sensor Networks - IPSN'04*, apr 2004, pp. 124 – 133.

[3] M. Rahimi, R. Pon, W. J. Kaiser, G. S. Sukhatme, D. Estrin, and M. Sirivastava, "Adaptive sampling for environmental robotics," in *IEEE International Conference on Robotics & Automation*, apr 2004, pp. 3537–3544.

[4] M. A. Batalin, M. Rahimi, Y.Yu, D.Liu, A.Kansal, G. Sukhatme, W. Kaiser, M.Hansen, G. J. Pottie, M. Srivastava, and D. Estrin, "Towards event-aware adaptive sampling using static and mobile nodes," Center for Embedded Networked Sensing - CENS, Tech. Rep. 38, 2004.

[5] G. P. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 51–58, may 2000.

[6] R. Nowak, U. Mitra, and R. Willett, "Estimating inhomogeneous fields using wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 6, pp. 999– 1006, 2004.

[7] M. Weiser and J. S. Brown, "The coming age of calm technolgy," in *Beyond calculation: the next fifty years*. Copernicus, 1997, pp. 75 – 85.

[8] GEORIO, *Alerta Rio*, Fundação Instituto de Geotécnica do Município do Rio de Janeiro, 2003, http://www2.rio.rj.gov.br/georio/site/alerta/quadro.asp.

[9] Y. Yu, D. Estrin, M. Rahimi, and R. Govindan, "Using more realistic data models to evaluate sensor network data processing algorithms," in *IEEE Workshop on Embedded Networked Sensors - EmNetS-I*, 2004.