In the Proceedings of the 2003 IEEE International Conference on Image Processing (ICIP). Special Session on Distributed Source Coding---*Invited Paper*.

SENSING LENA—MASSIVELY DISTRIBUTED COMPRESSION OF SENSOR IMAGES

Sergio D. Servetto

School of Electrical and Computer Engineering – Cornell University
http://people.ece.cornell.edu/servetto/

ABSTRACT

We consider the sensor broadcast problem: in our setup, sensors measure each one pixel of an image that unfolds over a field, and broadcast a rate constrained encoding of their measurements to every other sensor—the goal is for all sensors to form an estimate of the entire image. In recent work, we proposed a protocol that uses wavelets to decorrelate sensor data, taking advantage of the compact support of the basis functions to keep costly inter-sensor communication at a minimum. In this paper, we prove an asymptotic optimality result for these protocols: the rate of growth for the traffic they generate is $\Theta(\log(n/D))$ *(n nodes, total distortion D), matching* exactly *the rate of growth of the rate/distortion function. We thus close the gap between theory and practice for this new form of massively distributed (one pixel/sensor) image compression, by providing the first efficient and provably optimal algorithms to solve the sensor broadcast problem.*

1. INTRODUCTION

1.1. Problem Statement

Consider the following data transmission problem. n nodes v_{ij} are placed on a square grid of unit area, at locations $(x_i, y_i) = (\frac{i}{\sqrt{n}}, \frac{j}{\sqrt{n}})$, $1 \leq i, j \leq \sqrt{n}$ (for n large). Each v_{ij} observes *only one pixel* S_{ij} of an image defined over the field. This image is modeled as some spatial stochastic process with rate/distortion function $R_S(D)$, having the property that the correlation between samples increases as the distance between them in the grid decreases. Each v_{ij} wants to communicate an approximation of its S_{ij} to every other node in the network. Each v_{ij} can only send messages to and receive messages from its grid neighbors $v_{i-1,j}, v_{i,j-1}, v_{i+1,j}, v_{i,j+1}$, and these links have some finite capacity L. In the *sensor broadcast* problem, the goal is for each node v_{ij} to broadcast a rate constrained encoding of the sample observed to every other network location, so that collectively, all nodes can form an estimate of the image whose total distortion $\mathtt{E}(d(S, \hat{S})) < D$, for any prescribed value $D \geq 0$, for a distortion measure $d(\cdot, \cdot)$. This setup is illustrated in Fig. 1.

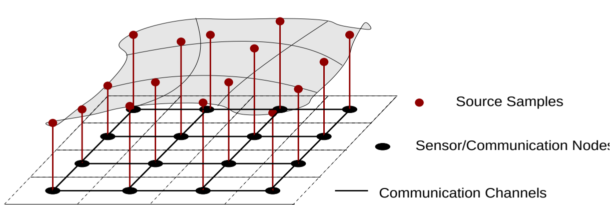

Fig. 1. Setup for the sensor broadcast problem. By having all nodes broadcast an encoding of their observations to every other node in the network, each node in the network is able to form an estimate of the entire image.

Work supported by the National Science Foundation, under awards CCR-0227676 and CCR-0238271 (CAREER).

In terms of applications, a most compelling one is the use of a sensor broadcast protocol as a building block in the construction of a distributed transmission array for the problem of *reachback* communication in sensor networks [1]. In this problem, the goal is to move the field of observations picked up by all sensors to a far receiver. What makes the reachback problem interesting is the fact that typically, each individual sensor does not have enough resources to generate a strong information bearing signal that can be detected reliably at the far receiver. A sensor broadcast protocol however allows all nodes to agree on a common stream of bits to send, and then all these nodes can synchronize their transmissions so as to generate a strong signal at the far point, a signal that results from coherently superimposing all the weak signals generated by each of the sensors [2]. This setup is illustrated in Fig. 2.

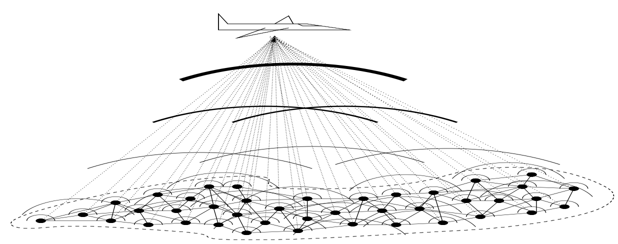

Fig. 2. Cooperation among sensors to reach back to a far receiver. A good analogy to describe the role of a sensor broadcast protocol in the context of reachback is that of a *symphonic orchestra*. When all instruments in the orchestra play independently, all we hear is noise; but when they all play according to a common script, the music from all instruments is combined into a coherent play. The bits distributed during sensor broadcast play the role of that common script for a later cooperative transmission step.

1.2. Main Contributions and Organization of the Paper

In previous work, we have presented the design of signal processing structures for the distributed computation of decorrelating transforms, to be applied on images generated by sensor arrays [6]. Decorrelation by means of a linear transform of the input, followed by bit allocation to an array of scalar quantizers operating on the transform coefficients, is a widely used data compression technique, whose rate/distortion performance is essentially optimal. The challenge however in the context of sensor networks lies in the fact that sensor measurements are not all available at a single location, but are distributed in the form of a single pixel per sensor. As a result, distributed algorithms are required to perform those computations. One such possible algorithm is shown in Table 1 (details on this algorithm, and an extension to 2D, can be found in [6]). The analysis of performance of this algorithm, both analytically and by means of numerical simulations, is the main contribution presented in this work.

In the rest of this paper, in Section 2 we carry out said performance analysis, and in Section 3 we offer concluding remarks.

```
SensorBroadcast (filters h_a g_a h_s g_s, filter length ℓ, node id ı)

for k ≥ 1 // index running over scales
  if (ı is of the form n2^k, for some n ∈ ℤ)
    { // this node has to carry out some computations
      Request samples < f(·), φ_{k−1}(· − 2^{−k+1} j) > (from subnets 2n ... 2n+ℓ−1 at
        scale k−1), and wait until all ℓ samples arrive;
      < f(·), φ_k(· − 2^{−k} n) > ← Σ_{j∈ℤ} h_a(2n − j) < f(·), φ_{k−1}(· − 2^{−k+1} j) >;
      < f(·), ψ_k(· − 2^{−k} n) > ← Σ_{j∈ℤ} g_a(2n − j) < f(·), φ_{k−1}(· − 2^{−k+1} j) >;
      Broadcast the new sample < f(·), ψ_k(· − 2^{−k} n) > to every other sensor;
      Send the new sample < f(·), φ_k(· − 2^{−k} n) > to sensor nodes at scale
        k + 1 which requested it;
    };
In parallel, listen to broadcasts from other sensors and record;
When enough data has arrived, use h_s and g_s to reconstruct the field.
```

Table 1. Pseudocode description for the algorithm executed at the $\imath$-th sensor. Note that each sensor will have multiple execution threads, and this is only one of them: network I/O for example, with functions used here, would be one such other thread.

2. PERFORMANCE EVALUATION

In this section we present results on the performance of the protocols from [6]. First we consider an asymptotic analysis based on sinc wavelets, then we present a simple estimate on energy consumption, and finally we present some numerical results.

2.1. Asymptotics Derived from the 1D Sinc Wavelet

2.1.1. Definitions and Problem Statement

Let X_t be a continuous time stationary Gaussian process, with mean function $\mu_X(t) = 0$, and autocorrelation function $R_X(\tau) \in \mathcal{L}^2(\mathbb{R})$. Following [4], we assume R_X is bandlimited with bandwidth W, i.e., $\hat{R}_X(f) = 0$ for $|f| > \frac{W}{2}$ ($\hat{R}_X$ is the Fourier transform of R_X). Let N be a (large) positive integer. We define a discrete time process $x_n = X_{n/N}$, consisting of samples of X taken at a distance of $\frac{1}{N}$: x_n has mean $\mu_x(n) = 0$, autocorrelation $R_x(m) = R_X(m/N)$, and power spectral density $\hat{R}_x(\omega) = N\hat{R}_X(N\omega)$. Consider now the sinc wavelet in discrete time, defined by the pair of filters

$$H_0(\omega) = \begin{cases} 1, & 0 \leq |\omega| < \frac{\pi}{2} \\ 0, & \frac{\pi}{2} \leq |\omega| \leq \pi \end{cases} \quad H_1(\omega) = \begin{cases} 0, & 0 \leq |\omega| < \frac{\pi}{2} \\ 1, & \frac{\pi}{2} \leq |\omega| \leq \pi \end{cases}$$

properly normalized. We implement it using a tree structured filter bank, in which an input signal is first filtered with both H_0 and H_1, then subsampled by a factor of 2, and then the process is iterated on the branch filtered with H_0. Let $Y_k(\omega)$ denote the frequency response of the system corresponding to the application of H_0+subsampling k times followed by one application of H_1+subsampling, and $Z_k(\omega)$ the frequency response corresponding to $k+1$ applications of H_0+subsampling. Then it is a trivial exercise to show that

$$\begin{aligned} Z_k(\omega) &= \begin{cases} 2^{\frac{k}{2}}, & 0 \leq |\omega| < \frac{\pi}{2^k} \\ 0, & \frac{\pi}{2^k} \leq |\omega| \leq \pi \end{cases} \\ Y_k(\omega) &= \begin{cases} 2^{\frac{k+1}{2}}, & \frac{\pi}{2^{k+1}} \leq |\omega| \leq \frac{\pi}{2^k} \\ 0, & \text{otherwise.} \end{cases} \end{aligned}$$

In a tree structured filter bank of depth K, an input signal with PSD $\hat{R}_x(\omega)$ is transformed into $K+1$ signals $\hat{R}_x(\omega)|Y_k(\omega)|^2$ ($k = 1...K$), and $\hat{R}_x(\omega)|Z_K(\omega)|^2$. Our goal next is to determine the minimum number of bits required to encode $\hat{R}_x(\omega)$, by encoding the resulting $K+1$ transform signals.

2.1.2. Rate Computation

Since R_X has bandwidth W, we have that $\hat{R}_x(\omega) = 0$ for $|\omega| > \frac{W}{2N}$. As a result, solving for k in $\frac{\pi}{2^{k+1}} = \frac{W}{2N}$, we find that for $k \leq \lfloor \log(\frac{\pi N}{W}) \rfloor$, the support sets of $\hat{R}_x(\omega)$ and $Y_k(\omega)$ are disjoint, and therefore that for all such k, $\hat{R}_x(\omega)|Y_k(\omega)|^2 = 0$, for all $-\pi \leq \omega \leq \pi$. This means that the first $\lfloor \log(\frac{\pi N}{W}) \rfloor$ branches of the filter bank contain no information, and hence no bits need be spent on the highpass projections.

Consider now $K = \lfloor \log(\frac{\pi N}{W}) \rfloor$. We have that $\hat{R}_x(\omega)|Z_K(\omega)|^2 = N\hat{R}_X(\omega N)2^K$, for $-\pi \leq \omega \leq \pi$. What is the minimum number of bits per sample required to encode this process? The answer is given by the rate/distortion function: a Gaussian process with PSD $\hat{S}(w)$ can be enconded with average per-sample distortion Δ using at least $R(\Delta)$ bits/sample, where

$$\begin{aligned} R(\Delta) &= \int_{-\pi}^{\pi} \frac{1}{2} \log\left(\frac{\hat{S}(\omega)}{\lambda_w}\right) d\omega \\ \lambda_w &= \begin{cases} \lambda, & 0 < \hat{S}(\omega) < \lambda \\ \hat{S}(\omega), & \hat{S}(\omega) \geq \lambda \end{cases} \\ \Delta &= \int_{\{\omega : \hat{S}(\omega) < \lambda\}} \hat{S}(\omega) d\omega. \end{aligned}$$

(This is the standard *reverse waterfilling* argument: we specify a target average distortion Δ, and then need to search for a threshold λ based on which to compute $R(\Delta)$.)

Next we note that, since for each sample of this filtered process there are 2^K samples of the input process, to achieve an average per-sample distortion d in the input process we must evaluate the rate/distortion function of this filtered process at a distortion $\Delta = d2^K$. So we obtain an estimate of λ from

$$\begin{aligned} d2^K &= \int_{\{\omega : \hat{R}_x(\omega) < \lambda\}} \hat{R}_x(\omega) d\omega \leq \int_{\{\omega : \hat{R}_x(\omega) < \lambda\}} \lambda d\omega \\ &= \lambda |\{\omega : \hat{R}_x(\omega) < \lambda\}| < \frac{\lambda W}{N}, \end{aligned}$$

and therefore we have that $\lambda > d2^K N/W$. Hence, defining $\nu =$

$\max_f \hat{R}_X(f) < \infty$, we have that

$$\begin{aligned} R(d2^K) &\leq \int_{-\frac{W}{2N}}^{\frac{W}{2N}} \frac{1}{2}\log\left(\frac{N\hat{R}_X(\omega N)2^K}{d2^K N/W}\right) d\omega \\ &\leq \frac{W}{N}\log\left(\frac{W\nu}{d}\right). \end{aligned}$$

2.1.3. Interpretation of the Rate Estimate

To avoid having to deal with boundary conditions, in the computation of the rate estimate above we worked with infinitely long sequences, and under that assumption, we obtained the average number of bits required to encode each sample. Suppose now we take a segment of finite length, and wlog, consider this segment to be the interval $[0, 1]$. In this case, we have a total of N samples, and the total rate required to encode them each with distortion $d = D/N$ (D is the total distortion in the N samples) does not exceed $N \cdot (\frac{W}{N}\log\left(\frac{W\nu}{d}\right)) = \Theta(\log(N/D))$.

Now, we claim that this establishes the asymptotic optimality of the proposed algorithms. We know that at high rates, the performance of a scalar quantizer followed by an entropy coder is within 0.255 bits of the rate/distortion bound (i.e., of the best performance achievable by vector quantizers of *any* length) [10]—that is, scalar processing of these filtered samples still achieves a rate of growth of $\Theta(\log(N/D))$. But we also know from [4] that the rate/distortion function of the entire field of measurements grows exactly like $\Theta(\log(N/D))$, and therefore, the proposed techniques are able to match that optimal rate of growth.

2.2. Energy Consumption Estimates

Due to the lack of actual sensor readings to work with, we chose to model sensor readings as a 2D stationary Gaussian process, with autocorrelation function $R(\tau) = (W/\pi)\,\mathrm{sinc}(W\tau)$ (where $\mathrm{sinc}(t) = \sin(t)/t$), and power spectral density $\hat{R}(f) = \mathrm{rect}\left(\frac{f}{2W}\right)$ (where $\mathrm{rect}(f) = 1$, if $|f| < \frac{1}{2}$, and $\mathrm{rect}(f) = 0$ otherwise). Therefore, our autocorrelation function has maximum bandwidth $2W$, where this bandwidth is measured in units of cycles/meter (instead of the classical temporal frequecies measured in units of Hz, or cycles/second). In Fig. 3, we show estimates of energy consumption for the transmission of each network snapshot.

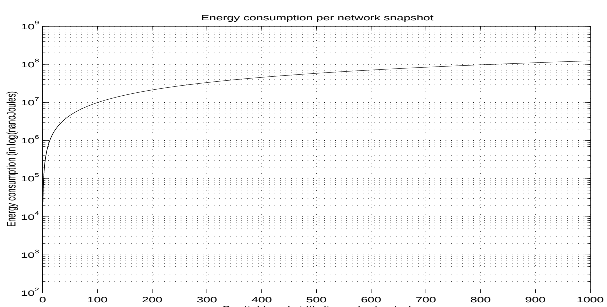

Fig. 3. Back-of-the-envelope estimates on energy consumption. In the horizontal axis we plot the spatial bandwidth W of the process being sensed; in the vertical axis, we plot the function $E_b \cdot W\log\left(\frac{WN}{D}\right)$, for an energy/bit $E_b = 7443$ nJ/bit (obtained from the specification of an experimental UWB radio interface), number of nodes N, and average distortion $D/N = 0.01$ (a number much smaller than the spatial bandwidth W, resulting in negligible distortion in the reconstructions obtained), for $0 \leq W \leq 1000$.

Note two important aspects of Fig. 3. First, whereas the value of E_b corresponds to actual hardware, the choice of W clearly depends on the statistics of the measurements to be picked up by this network. Lacking real measurements, all we can do is pick a reasonable range of values for this spatial bandwidth, and look at how the amount of traffic generated by our network would scale—the range of $0 \leq W \leq 1000$ is chosen on the basis that, for $W = 1000$, we are considering spatial processes with fluctuations in the order of 1mm, and this appears to us more than enough to capture phenomena like temperature / light / seismic activity / etc. Note also that a most important result that follows from theoretical developments is that the amount of traffic depends only on the *average* mean-squared error and on the spatial bandwidth W, but not on the number of nodes in the network: more nodes in the network generate more data, but this data will be more correlated, and therefore can be compressed more. This is the key intuition that all of our ideas about joint routing and compression are built upon.

Basically, the plot above shows that it should be possible to sense a spatial process with fluctuations of duration in the order of 1mm, replicate a snapshot of the measurements collected by the entire network at all nodes, and consume approximately $10^8 nJ$, or about 0.1 Joule per snapshot at each node participating in the broadcast operation. This number appears well within the resources of existing batteries—for example, browsing on the web we found one commercially available NiCd battery capable of storing approximately 500 Joules.

2.3. Numerical Results

We complement the performance analysis outlined above with some numerical simulations. The algorithms described above have been implemented both in MATLAB and in C. The simulator in C implements exactly the algorithm of Table 1 (in two dimensions), for a square grid network. This simulation includes detailed models of internal buffers in the network, generates traces of which coefficients are sent over which link at which point in time, etc., but works only for Haar filters. The MATLAB simulation works with any filters in the wavelet toolbox, but does not implement all the detailed network functions implemented in C: it simply does a centralized computation of the wavelet coefficients, quantizes them, and then picks one particular link in the square grid and counts how many bits would go through that link. The rationale for working with two simulators is that there are aspects which are much easier to code up in C (network simulation), whereas there are others which are much easier to code up in MATLAB (wavelet processing). Yet an important piece of evidence that points to the correctness of *both* implementations is the fact that, when using the MATLAB simulator with Haar wavelets, the traffic loads obtained match *exactly* those of the C simulator. A plot of the resulting traffic loads is shown in Fig. 4.

3. CONCLUSIONS

In this paper we have continued with work started in [4, 5, 6], on the compression of data collected by dense sensor networks. We have shown in this work how wavelets can effectively decorrelate sensor data under decentralization constraints, via both analysis and numerical simulations.

3.1. Some Comments on the Work of Marco et al. [12]

In recent work, Marco et al. [12] expressed doubts about the correctness of some of our results (joint with A. Scaglione) published in [4]. Since those are closely related to the results presented in this paper, we feel it is appropriate to offer some clarification.

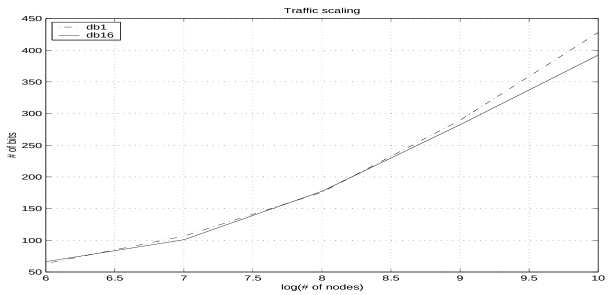

Fig. 4. Traffic loads based on two different Daubechies wavelets: `db1` (the Haar wavelet), and `db16`, for the Gaussian process described above, with a sampling interval of $\frac{1}{2^k}$—k is the horizontal axis, and the number of bits generated is the vertical axis. Observe how, in the semilogx plot, the curve is nearly a straight line, thus suggesting the log behavior predicted by theory. Observe also that as we increase the number of zeros at π in the filter, the traffic load appears to become a straight line more closely.

In [4], we established the following:

- There exist joint routing and data compression strategies capable of generating a total of $O(\log(n/D))$ bits/snapshot.
- Under some conditions, the rate/distortion function of the whole network grows as $O(\log(n/D))$ bits/snapshot.

From these results we derived a number of conclusions, both of which (results and conclusions) were questioned in [12].

One inference we made is that it should be possible for any node in a sensor network to obtain an estimate of the entire field of measurements, within any prescribed distortion value, for asymptotically large networks. If we observe the asymptotic form for the rate of growth of the rate/distortion function we note that, provided the average per/sample distortion $\frac{D}{n} = \Theta(1)$ (i.e., that it is maintained constant), the total number of bits generated is constant too. Therefore, as we claimed in [4], assuming a constant per-sample distortion, the size of a snapshot does not depend on the size of the network: as more nodes are added into the network, more data is generated, but this data is more correlated and it can be compressed more. It is not correct though that *any* total distortion D can be achieved, as we said in [4]. It follows from the max-flow/min-cut theorem that each node can receive only $\Theta(1)$ bits, since that is the capacity of a cut with a receiver on one side, and all other nodes on the other side. Such a cut can easily be shown to be a minimum cut—for reference see, e.g., [7, Ch. 26]; using a different type of argument, a similar claim was made in [12, pp. 10–15]. Therefore, our statement should have been that a distortion D in a network of size n can be achieved, for all n, but only provided that the capacity of the individual network links $\ell = \Omega(\log(n/D))$.

Another aspect of our work doubted in [12] refers to our use of a definition of capacity which does not involve any notion of time. We would like to point out that there exist notions of capacity in the networking literature which do not necessarily involve time. One such example, not mentioned explicitly in [4] (and we regret having assumed this was obvious, thus omitting appropriate references) is the notion of *maximum stable throughput* [13]—in the context of our work, we later expanded significantly along these lines in [3]. We say that a throughput of $\Theta(f(n))$ is stable if, for an aggregate traffic generated by all sources of $\Theta(f(n))$ bits, the size of the longest internal queue in the network remains bounded. As showed in [4], n independent encoders generate snapshots of size $\Theta(n\log(n))$ bits, for any distortion D (D is a constant hidden by the big-oh notation). As a result, it is not difficult to show that there are buffers in the network whose size grows as $\Theta(\log(n))$, and therefore that for any rate $\epsilon > 0$, the injection of ϵ bits at each node results in a total throughput that is *not* stable. On the other hand, with the joint routing and compression techniques whose existence was established in [4], and actually constructed in this paper, such instabilities are eliminated.

One more aspect doubted in [12] is whether the use of the rate/distortion function as a measure of how much data is generated by the sensor network is appropriate. This doubt is based on the fact that to approximate the rate/distortion function we require processing of blocks of arbitrarily large size, whereas in a sensor network we are only allowed to perform scalar processing. In this regard, we would like to highlight an old result of information theory which states that, at high rates and for Gaussian sources, the performance gap between an optimal vector quantizer (capable of approximating the r/d function arbitrarily closely) and a scalar quantizer followed by an entropy coder is 0.255 bits/sample, irrespective of the dimension of the vector quantizer [10, pg. 2333]. Therefore, the rate of growth in the number of bits generated by scalar processing differs from that of the rate/distortion function by only a constant factor, and hence are the same in the big-oh sense, thus making all the claims in [4] related to this topic perfectly valid.

In summary: (a) we maintain that the sensor broadcast problem can be solved for *any* network size, provided $\frac{D}{n} = \Theta(1)$; (b) we stand corrected on our claim that any distortion $D > 0$ is achievable in the sensor broadcast problem—only distortions for which $\ell = \Omega(\log(n/D))$ are indeed achievable; (c) we maintain that it is possible to define meaningful notions of capacity without involving time, based on which our claims in [4] are perfectly valid; (d) we maintain that it is perfectly valid to measure the rate of growth of the total traffic generated by the network using the rate/distortion function of the entire network, as done in [4].

3.2. Future Work

There are a number of topics that we will need to address for the writeup of a journal paper along these lines: (a) extend our results to random networks, instead of the regular sampling patterns considered in this work; (b) extend the analysis of Section 2 to 2D (but not the simulators, these do work on 2D grids already); and (c) repeat the analysis of Section 2 for the family of maxflat filters (used in the simulations), to determine the rate of convergence of the traffic loads to $\Theta(\log(N/D))$ as a function of the number of zeros at π of the filters.

Acknowledgement—The author would like to thank Jason K.-C. Chen (a student in the MEng program at Cornell), for help in the development of the numerical simulations.

4. REFERENCES

[1] J. Barros and S. D. Servetto, "On the Capacity of the Reachback Channel in Wireless Sensor Networks," in *Proc. IEEE Int. Workshop Multimedia Sig. Proc.*, US Virgin Islands, 2002, Invited paper to the special session on *Signal Processing for Wireless Networks*.

[2] A. Hu and S. D. Servetto, "Optimal Detection for a Distributed Transmission Array," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, Yokohama, Japan, 2003.

[3] C. Peraki and S. D. Servetto, "On the Maximum Stable Throughput Problem in Random Networks with Directional Antennas," in *Proc. ACM MobiHoc*, Annapolis, MD, 2003.

[4] A. Scaglione and S. D. Servetto, "On the Interdependence of Routing and Data Compression in Multi-Hop Sensor Networks," 2003, ACM/Kluwer Mobile Networks and Applications. Special issue with selected papers from ACM MobiCom 2002. To appear. Available from `http://people.ece.cornell.edu/servetto/`.

[5] S. D. Servetto, "Quantization with Side Information: Lattice Codes, Asymptotics, and Applications in Wireless Networks," Submitted to the IEEE Trans. Inform. Theory. Available from `http://people.ece.cornell.edu/servetto/`.

[6] S. D. Servetto, "Distributed Signal Processing Algorithms for the Sensor Broadcast Problem," in *Proc. 37th Annual Conf. Inform. Sciences Syst. (CISS)*, Baltimore, MD, 2003.

[7] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms (2nd ed)*, MIT Press, 2001.

[8] I. Daubechies, *Ten Lectures on Wavelets*, SIAM, 1992.

[9] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks," *IEEE Trans. Inform. Theory*, vol. 46, no. 2, pp. 388–404, 2000.

[10] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Trans. Inform. Theory*, vol. 44, no. 6, pp. 2325–2383, 1998.

[11] U. Grenander and G. Szegö, *Toeplitz Forms and their Applications*, University of California Press, 1958.

[12] D. Marco, E. J. Duarte-Melo, M. Liu, and D. L. Neuhoff, "On the Many-to-One Transport Capacity of a Dense Wireless Network and the Compressibility of its Data," in *Proc. Int. Wkshp. Inform. Proc. Sensor Networks*, 2003.

[13] B. S. Tsybakov and V. A. Mikhailov, "Ergodicity of a Slotted ALOHA System," *Problemy Peredachi Informatsii*, vol. 15, no. 4, pp. 301–312, 1979.