ENERGY EFFICIENT SOURCE LOCALIZATION ON A MANHATTAN GRID WIRELESS SENSOR NETWORK

Matthew A. Prelee and David L. Neuhoff

EECS Department, University of Michigan, Ann Arbor, MI 48109

ABSTRACT

For wireless sensor networks, many decentralized algorithms have been developed to address the problem of locating a source that emits acoustic or electromagnetic waves based on received signal strength. Among the motivations for decentralized algorithms is that they reduce the number of transmissions between sensors, thereby increasing sensor battery life. Whereas most such algorithms are designed for arbitrary sensor placements, such as random placements, this paper focuses on applications that permit a choice of sensor placement. In particular, to make communications costs small, it is proposed to place sensors uniformly along evenly spaced rows and columns, i.e., a Manhattan grid. For such a placement, the *Midpoint Algorithm* is proposed, which is a simple noniterative decentralized algorithm. The results of this paper show that Manhattan grid networks offer improved accuracy vs. energy tradeoff over randomly distributed networks. Results also show the proposed Midpoint Algorithm offers further energy savings over the recent POCS algorithm.

Index Terms— Wireless sensor networks.

1. INTRODUCTION

An important application in the area of wireless sensor networks is the problem of source localization, where sensor measurements are used to estimate the position of a source emitting electromagnetic or acoustic waves [1]. One such measurement is received signal strength (RSS), where noisy measurements of received signal power are made at each sensor. Many previous RSS-based localization algorithms for a single source are designed for arbitrary sensor placements [2, 3, 4, 5]. While it is true that some applications may not allow for a choice in sensor placement, those that do have certain advantages. For such, we propose placing sensors uniformly along rows and columns spaced k sensors apart, as illustrated in Figure 1. As we will argue, this *Manhattan grid* sensor layout has the advantage that communication requires less power than a randomly distributed network or uniform lattice, thereby increasing battery life. Previously, Manhattan grid sampling has been used in bilevel image coding and reconstruction [6, 7, 8], as well as grayscale image reconstruction [9, 10]. A sampling theorem for Manhattan grids has also been derived [11]. The problem of RSS source localization is similar to these image processing problems in that we are given noisy samples of an RSS "image" on the Manhattan grid and we would like to reconstruct properties of that image (in this case, the source location).

We begin by describing geometric properties of a Manhattan grid. An infinite Manhattan grid is described by two parameters: its intersensor spacing λ and the Manhattan grid parameter k. This grid partitions space into $k\lambda \times k\lambda$ blocks. Associating $2k-1$ sensors with each block, we see that the sensor density is $\frac{2k-1}{(k\lambda)^2}$. Now suppose we would like to construct a finite Manhattan grid with n sensors and parameter k over a $w \times w$ square region. Since for some values of k it is impossible to choose λ so that a Manhattan grid with n sensors and parameters (k, λ) exactly covers the $w \times w$ square, we choose sensor spacing $\lambda = w\sqrt{(2k-1)/(nk^2)}$, and generate a finite Manhattan grid with $B = \left(\lfloor \frac{w}{k\lambda} \rfloor\right)^2$ grid blocks and $(2k-1)B + 2k\sqrt{B} - 1 \approx n$ sensors.

This work was supported by NSF grant CCF 0830438.

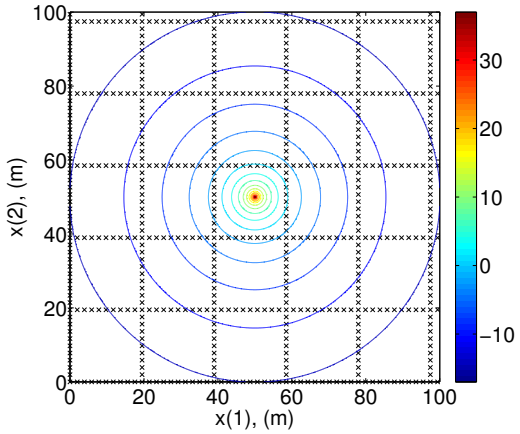

Fig. 1. $n = 500$ sensors placed along a Manhattan Grid ($k = 10$) in a 100m $\times$ 100m square; each $\times$ denotes one sensor. A source is located at $\theta = [50, 50]$. Contours of constant power under no noise are shown in dB.

We now describe the energy cost advantages of a Manhattan grid sensor network, and later we propose the *Midpoint Algorithm* for further reductions in energy. Mimicking the analysis in [3, 12], the total energy needed for any source localization algorithm is

$$\mathcal{E} = b \times h \times e \tag{1}$$

where b is the number of total sensor transmissions, h is the number of hops through the network per transmission, and e is the energy required to transmit a single hop. As we will now discuss, e is greatly affected by the choice of sensor placement, in particular the intersensor spacings, and since for typical algorithms h and b are not nearly as affected, a first-order approach for comparing the energy performance of various sensor layouts is to estimate e. To do so, note that the average power emitted by a sensor at distance d decays as $\mathcal{O}(d^{-\alpha})$, where α is between 2 and 4. Thus, the energy required for one communication hop between neighboring sensors is $e = \mathcal{O}(d^{\alpha})$. If we place n sensors randomly over a $w \times w$ square, the average distance between neighboring sensors is $w/\sqrt{n}$ and $e = \mathcal{O}(n^{-\alpha/2})$. However, when the sensors are placed with spacing λ along a Manhattan grid of parameter k and density n/w^2, the distance between neighboring sensors is $\lambda = w\sqrt{(2k-1)/nk^2}$, and $e = \mathcal{O}(k^{-\alpha/2}n^{-\alpha/2})$, thus reducing energy by a factor $\mathcal{O}(k^{\alpha/2})$ over a random placement.

We formulate the problem in Section 2. In Section 3, the decentralized *Midpoint Algorithm* is proposed to solve the source localization problem on a Manhattan grid. The accuracy vs. energy performance of the Midpoint Algorithm is discussed in Section 4 and compared to the recent decentralized POCS algorithm [4]. Finally, we discuss avenues for future research in Section 5.

2. PROBLEM STATEMENT

Suppose n sensors are distributed over a $w \times w$ square along a Manhattan grid. Along each row/column of the Manhattan grid, $m = w/\lambda$ sensors are spaced $\lambda = w\sqrt{(2k-1)/nk^2}$ apart. For some positive integer k, each row/column of the grid is spaced $k\lambda$ apart. These four quantities n, m, k, λ are dependent, so we fix the number of sensors n and vary the Manhattan grid parameter k.

Let $x_i = [x_i(1),\, x_i(2)]$ denote the location of the ith sensor. A source with known intensity A is positioned at unknown location $\theta = [\theta(1),\ \theta(2)]$ within the $w \times w$ square. The source emits a signal whose strength decays with distance to the power β, where β is typically in the range of 2 to 4. For each i, the ith sensor makes a noisy measurement y_i of the received signal strength (RSS), modeled as

$$y_i = \frac{A}{\|x_i - \theta\|^\beta} + v_i. \tag{2}$$

where $\|\cdot\|$ is the Euclidean norm and the v_i's are i.i.d. zero-mean Gaussian noise with known variance σ^2. [4] contains more information about the theory behind this model, and an application using real-world data can be found in [13]. Extensions to noise models involving fading [1] are possible.

Depending upon the sensor deployment geometry and localization algorithm, groups of neighboring sensors must communicate with each other and decide if the source is within their vicinity. If so, they must also estimate its location. Hence, our proposed algorithm must operate in a decentralized manner. A decentralized algorithm's performance is determined by (a) the probability that sensors locally close to the source will actually detect the source, (b) the false alarm probability, i.e., the probability that a sensor located far from the source will mistakenly declare a detection, and (c) the average squared error between the true location θ and its estimate $\widehat{\theta}$ in cases of a correct detection. We also consider the communication cost (1).

In [3], Rabbat and Nowak proposed a decentralized source localization algorithm based on incremental subgradient optimization, but they did not specify how to detect the presence of a source before making an estimate. In [4], Blatt and Hero proposed a decentralized source localization method based on projections onto convex sets (POCS). This algorithm required choosing a threshold γ such that all sensors with received RSS greater than γ were considered *active*; thus, the detection probability was determined by this threshold. The active sensors collaborated to produce an estimate of θ based on their RSS measurements. One improvement of POCS over Rabbat and Nowak's method was a smaller energy cost (1); in particular, POCS reduced the number of sensor transmissions b required for their algorithm to converge. Both iterative algorithms can be applied to a Manhattan grid sensor network to solve the problem of source localization. However, we will propose a *non-iterative* algorithm that exploits the Manhattan grid geometry to reduce communication costs at the price of higher estimation error.

Finally, in [5], Rabbat and Nowak consider various estimators to solve the problem of source localization, including

$$\widehat{\theta} = \frac{\sum_{i=1}^n x_i 1_{\{y_i > \gamma\}}}{\sum_{i=1}^n 1_{\{y_i > \gamma\}}} \tag{3}$$

where γ is a threshold and $1_{\{y_i>\gamma\}}$ is the indicator function. This estimator is simply an average of active sensor locations, and we modify it for a Manhattan grid in the next section.

3. MIDPOINT ALGORITHM

In order to take advantage of the Manhattan grid structure to reduce the communication cost of forming an estimate of θ, the *Midpoint Algorithm* differs from the (3) in two principal ways. First, instead of jointly estimating $\theta(1)$ and $\theta(2)$ from all active sensors, the Midpoint Algorithm estimates $\theta(1)$ from active sensors in Manhattan grid rows, and $\theta(2)$ from active sensors in grid columns. Second, it replaces the average location in (3) with the midpoint between the active sensors in each grid row (grid column) that are farthest apart.

For concreteness, denote the set of sensors in the jth row by

$$\mathcal{H}_j = \{i : x_i(1) = \ell\lambda,\ x_i(2) = jk\lambda,\ \ell = 0, \cdots, m-1\}.$$

We say that sensor i in $\mathcal{H}_j$ is *active* if $y_i > \gamma$, and we say that the jth row $\mathcal{H}_j$ is *active* if it contains an active sensor. Whereas one could estimate $\theta(1)$ from an active row as in (3):

$$\widehat{\theta}_j(1) = \frac{\sum_{i\in\mathcal{H}_j} x_i(1) \cdot 1_{\{y_i>\gamma\}}}{\sum_{i\in\mathcal{H}_j} 1_{\{y_i>\gamma\}}},$$

the Midpoint Algorithm estimates $\theta(1)$ as

$$\widehat{\theta}_j(1) = \frac{x_a(1) + x_b(1)}{2} \tag{4}$$

which is simply the midpoint of the first coordinates of the left- and rightmost active sensors x_a and x_b in grid row j. These will be called *endpoints* of the active row. Similarly, from sensors in grid columns, an estimate $\widehat{\theta}(2)$ of $\theta(2)$ is made for every active column. If there is at least one active row and one active column, then an estimate $\widehat{\theta} = (\widehat{\theta}(1), \widehat{\theta}(2))$ can be made for each pairing of an active row and active column. For each such pair, the corner point shared by the row and column is called a *decision corner*.

In Section 3.1 it will be shown how to choose γ to ensure that with high probability at least one of the four corners of the $k\lambda \times k\lambda$ block containing the source is a decision corner and there are no decision corners outside the block. Section 3.2 describes a distributed communication protocol that distributes endpoint locations so as to (1) enable those corners lying on the aforementioned block to determine whether or not they are decision corners, and (2) to enable such decision corners to make their estimates of θ.

Note that in the absence of noise, sensors in a grid row (column) will lie in a single consecutive interval. Although this does not necessarily happen in the presence of noise, results in Section 4 show that the Midpoint Algorithm works well nevertheless.

3.1. Choosing the threshold

Our choice of γ heavily impacts the performance of the Midpoint Algorithm. If γ is too large, then the probability of having at least one decision corner will not be large, i.e., the probability of missed detection will be too large. If γ is too small, there will be spurious decision corners, leading to high probability of false alarms and poor estimates of θ. Thus the goal of this section is to find an upper bound γ_1 and lower bound γ_2 such that the these undesirable events occur with low probability for any threshold satisfying $\gamma_2 \le \gamma \le \gamma_1$.

We begin with the upper bound γ_1. Let E_1 be the event that at least one of the corners of the $k\lambda \times k\lambda$ block containing the source is a decision corner, thereby indicating a successful detection. We

want to choose γ small enough that the $\Pr(E_1) \geq 1 - \varepsilon_1$, where ε_1 is some small tolerance. A useful fact is that the closest row sensor and the closest column sensor to the source are within distance $\frac{\lambda}{2}\sqrt{k^2+1}$. If these sensors are active, then E_1 occurs. Therefore, using this fact and the union bound, it can be shown that

$$\Pr(E_1) \geq 1 - 2\Pr\left(\frac{A}{(\frac{\lambda}{2}\sqrt{k^2+1})^\beta} + v \leq \gamma\right).$$

Since v is Gaussian with known variance σ, equating the RHS of the above to $1 - \varepsilon_1$ yields the fact that if

$$\gamma \leq \gamma_1 \triangleq \frac{2^\beta A}{(\lambda\sqrt{k^2+1})^\beta} - \sigma\mathcal{Q}^{-1}\left(\frac{\varepsilon_1}{2}\right), \tag{5}$$

then $\Pr(E_1) \geq 1 - \varepsilon_1$. In the above, $\mathcal{Q}(x) = \Pr(X > x)$ for a zero mean, unit variance Gaussian random variable X, and $\mathcal{Q}^{-1}$ is its inverse function.

In a similar manner, we calculate a threshold lower bound γ_2. Consider the event E_2 that there are no decision corners outside the $k\lambda \times k\lambda$ block containing the source, so there are at most four decisions corners. We want to choose γ so that $\Pr(E_2) \geq 1 - \varepsilon_2$. This also ensures that the false alarm rate will be at most ε_2. Using the fact that E_2 occurs when all sensors farther than $k\lambda$ from the source are inactive, it can be shown using the union bound that

$$\Pr(E_2) \geq 1 - n\Pr\left(\frac{A}{(k\lambda)^\beta} + v > \gamma\right).$$

Again, equating the RHS of the above event to $1 - \varepsilon_2$ yields the fact that if

$$\gamma \geq \gamma_2 \triangleq \frac{A}{(\lambda k)^\beta} + \sigma\mathcal{Q}^{-1}\left(\frac{\varepsilon_2}{n}\right), \tag{6}$$

then $\Pr(E_2) \geq 1 - \varepsilon_2$.

For large k values, the Manhattan grid "block size" $k\lambda$ becomes very large, and it becomes physically impossible to detect certain source locations without incurring a large false alarm rate. Thus, in our experiments, we only choose values of k small enough that $\gamma_1 > \gamma_2$. We found that the Midpoint Algorithm performs better for large γ, so we set our threshold to be the upper bound $\gamma = \gamma_1$.

3.2. Communication protocol and costs

By our choice of γ in the previous section, with high probability there will be at least one decision corner on the $k\lambda \times k\lambda$ block containing the source, and there will be no decision corners outside this block. We now describe a distributed communication protocol by which sensors efficiently report endpoint data to corner points, enabling those that are decision corners to recognize that they are such and to make their estimates.

Assume sensor clocks are synchronized. Time is slotted and the system operates with cycles of $8m$ slots, where m is the number of sensors in a row or column. The following protocol operates during the first $4m$ slots along rows, and repeats during the next $4m$ slots along columns.

During the first m time slots, messages are sent left-to-right across each row of the grid. Specifically, during slot t, only sensor t of each row may transmit, and neighboring sensor $t+1$ listens[1]. If sensor t of row j did not hear a message (from $t-1$) during the previous time slot, it knows the first active sensor in its grid row has not been found (the first endpoint). It then compares its measured RSS to the threshold. If $y_t < \gamma$, sensor t is not active and does not transmit. However, if $y_t > \gamma$, sensor t is active and transmits 0 to its neighbor $t+1$, thereby marking t as the first endpoint.

If, on the other hand, sensor t did hear a message from $t-1$, it increments the message by 1 and transmits the new message to sensor $t+1$. Thus, each message is an integer representing the distance to the first active sensor in the row. Message-passing ends after the message is received by two corner sensors (we assume sensors know whether they are placed on a corner *a priori*). This requires an extra "corner counting" bit to be sent along with each transmission.

During the next m time slots, the sensor order is reversed and messages are passed right-to-left in a similar manner in order to determine the second endpoint in the row. In some cases, after these $2m$ time slots, at least one corner knows the locations of both endpoints and can estimate $\theta(1)$. However, if both endpoints are less than $k-2$ sensors apart, it is possible that the endpoints lie entirely between two adjacent corners, and these corners will only know one endpoint apiece. In this case, the two endpoints (and the sensors in-between) will know both endpoint locations. Thus, the next $2m$ time slots are reserved for the endpoints to transmit the missing endpoint locations to their closest corner sensor.

As mentioned earlier, this protocol is repeated for columns in the next $4m$ time slots. After all $8m$ time slots, any corner that has received both horizontal and vertical pairs of endpoints, recognizes itself as a decision corner and makes an estimate $\widehat{\theta}$. It can be seen that this protocol finds at least one decision corner on the block containing the source, if there is one, which happens with high probability.

We now find an upper bound to the communication costs of the protocol. Due to our choice of threshold, it is easy to see that each decision corner will be within $2k$ sensors of an endpoint with probability $1 - \varepsilon_2$. It can be shown that this protocol requires at most $4k$ transmissions per active row. Each distance transmission requires $\lceil \log_2(2k) \rceil$ bits to transmit endpoint data with an overhead of 1 bit for corner counting, totaling $2 + \lceil \log_2 k \rceil$ bits per transmission.

4. EXPERIMENTS AND RESULTS

For our experiments, we chose $w = 1000$, $n = 10,000$, and for various values of k we designed finite Manhattan grids as described in the introduction. We set $A = 10,000$, $\sigma = 1$, $\beta = 2$, and the threshold γ was set to the upper bound $\gamma = \gamma_1$ with $\varepsilon_1 = \varepsilon_2 = 10^{-5}$. To avoid edge effects, θ was distributed randomly in a $k\lambda \times k\lambda$ block near the center of the 1000 m $\times$1000 m square. We tested $k = 2, \ldots, 14$, all of which satisfied the condition $\gamma_1 > \gamma_2$. For each value of k, the squared error was calculated for estimates produced by both the Midpoint Algorithm and the POCS algorithm [4]. 20,000 trials of this experiment were performed, during which we did not observe any missed detections or false alarms for either algorithm. In some trials, multiple estimates of θ were generated by the Midpoint Algorithm using different decision corners. The choice of these multiple estimates did not impact the overall performance of our algorithm, so we chose an estimate randomly.

The POCS algorithm was chosen for comparison because of its high accuracy and low communication costs, needing many fewer cycles to converge than other algorithms such as [3]. POCS also required a choice of threshold γ_{POCS}; we observed that POCS performed better when a slightly smaller threshold was used than the Midpoint Algorithm. To still ensure a detection probability of at least $1 - \varepsilon_1$ and a false alarm rate less than ε_2, we chose $\gamma_{POCS} = \gamma_2$. The POCS algorithm also required a convergence threshold; we used the value of 10^{-3} as used in [4]. In addition to running POCS on

[1]Sensors in adjacent Manhattan grid rows are presumed to be far enough away (at least $k\lambda$) that transmissions from adjacent grid rows do not interfere.

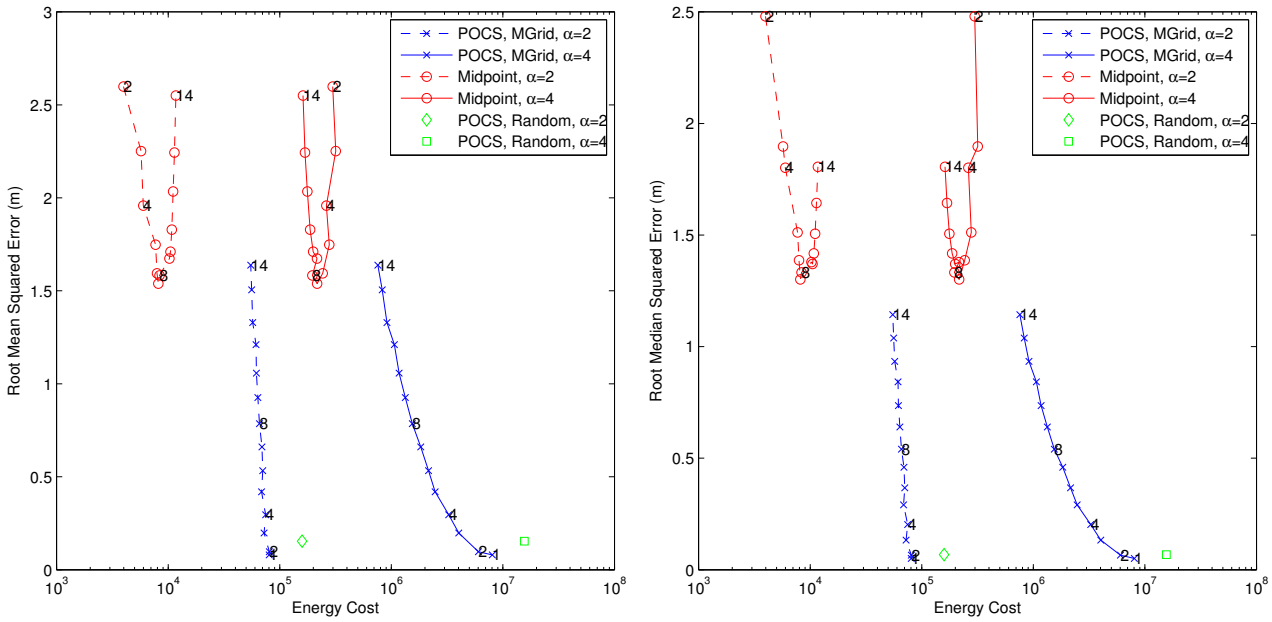

Fig. 2. Accuracy vs. energy cost tradeoff for our proposed Midpoint Algorithm and the POCS algorithm [4] for $\alpha = 2$ and $\alpha = 4$. Values of k are labeled for some points. POCS was also run on a uniform lattice (labeled $k = 1$) and a randomly distributed network. RMSE vs. energy is shown in (a) and RMedSE vs. energy cost is shown in (b).

a Manhattan grid, we ran POCS for sensors placed on a uniform lattice (labeled $k = 1$) as well as for randomly placed sensors. For these experiments, we found that thresholds of $\gamma_{lattice} = 360$ and $\gamma_{random} = 15$ worked well (for comparison, $\gamma_2 = 106$ with $k = 1, \varepsilon_1 = \varepsilon_2 = 10^{-5}$). Note that we need a much smaller threshold for the random network because, unlike the uniform lattice, we are not guaranteed to have a sensor close to the source.

In addition, bounds on the energy cost of each algorithm were calculated. Suppose there are r active sensors above threshold. The POCS algorithm needs some number of cycles c to converge and one extra cycle to calculate an average estimate of θ. Note that c typically depends on some convergence criteria; we used the default criteria suggested in [4], which was that the previous estimate of θ during the last cycle is within 10^{-3} of the new estimate for the first sensor in the cycle. For our experiments, c ranged between 4 and 7. Although we used double precision for $\widehat{\theta}$ in our simulation, we assumed that each coordinate of θ was quantized to 3 significant decimal places when being transmitted, which corresponds to $\lceil \log_2(10^3) \rceil = 10$ bits per coordinate. Thus, $b_{POCS} = 20 \cdot (c+1) \cdot r$. Finally, we conservatively assumed that $h = 1$ for POCS since *most* transmissions are between neighbors. This is conservative because some transmissions require inactive sensors to relay data between active sensors, in which case $h > 1$. Thus, for each trial we calculated

$$\mathcal{E}_{POCS} = 20r(c+1)\lambda^{\alpha}.$$

Note that this is a conservative lower bound on the true energy.

Now we consider the energy cost of the Midpoint Algorithm. Define N_{row} and N_{col} to be the number of active rows and columns, respectively. Following the discussion in Section 3.2, at most $4k$ transmission are needed per active row/column, and we transmit $2 + \lceil \log_2 k \rceil$ bits per transmission. Each transmission is always between neighboring sensors, so unlike the POCS algorithm, we always have that $h = 1$. Thus, the total energy required is at most

$$\mathcal{E}_{midpoint} = 4k(2 + \lceil \log_2 k \rceil)(N_{row} + N_{col})\lambda^{\alpha}.$$

Observe that the $4k$ transmissions per active row/column is an upper bound on the number of transmissions. We emphasize that $\mathcal{E}_{midpoint}$ is a conservative upper bound for the Midpoint Algorithm, whereas $\mathcal{E}_{POCS}$ is a conservative lower bound for POCS. These energy cost bounds were calculated for both $\alpha = 2$ and $\alpha = 4$. We plotted both the root mean squared error (RMSE) and root median squared error (RMedSE) vs. energy cost in Figure 2. Plotting the root median squared error is useful because of its insensitivity to outliers.

First, we consider the performance of POCS for either value of α. When POCS was run on a uniform lattice and Manhattan grid, less energy was used than on a randomly distributed lattice. However, error also gradually increased as k increased. This shows the fundamental tradeoff between a random network, a uniform lattice network, and a Manhattan grid. That is, if we are willing to tolerate an increase in error, the Manhattan grid requires much less energy.

Now let us compare the performance of the Midpoint Algorithm to POCS. If we are willing to sacrifice more accuracy, the Midpoint Algorithm uses even less energy than POCS for all values of k and fixed α. For a fixed accuracy level, it is possible to make POCS more competitive by choosing a convergence threshold larger than 10^{-3}, thereby reducing the required number of transmissions while decreasing the accuracy. However, even when we increased this threshold, we found that the Midpoint Algorithm outperformed POCS for a fixed achievable accuracy. It is interesting to point out that for k increasing and n fixed, the energy cost of the Midpoint Algorithm increases for $\alpha = 2$ and decreases for $\alpha = 4$. Note that $\mathcal{E}_{midpoint} = \mathcal{O}(k^{1-\alpha/2} \log k)$. Thus, the energy cost increases as $\mathcal{O}(\log k)$ for $\alpha = 2$, but decreases as $\mathcal{O}(\log(k)/k)$ when $\alpha = 4$.

5. FUTURE WORK

There is an opportunity in future work to extend the Midpoint Algorithm to the case of unknown source and/or noise power. There may also be other nonrandom sensor layouts that yield better tradeoffs between accuracy and communication cost than a Manhattan grid.

6. REFERENCES

[1] N. Patwari, J.N. Ash, S. Kyperountas, A.O. Hero III, R.L. Moses, and N.S. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Sig. Proc. Magazine*, vol. 22, no. 4, pp. 54–69, 2005.

[2] X. Sheng and Y.H. Hu, "Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks," *IEEE Trans. Sig. Proc.*, vol. 53, no. 1, pp. 44–53, 2005.

[3] M.G. Rabbat and R.D. Nowak, "Decentralized source localization and tracking," in *IEEE ICASSP*, 2004, vol. 3, pp. iii–921.

[4] D. Blatt and A.O. Hero, "Energy-based sensor network source localization via projection onto convex sets," *IEEE Trans. Sig. Proc.*, vol. 54, no. 9, pp. 3614–3619, 2006.

[5] M. Rabbat, R. Nowak, and J. Bucklew, "Robust decentralized source localization via averaging," in *IEEE ICASSP*, 2005, vol. 5, pp. v–1057.

[6] M.G. Reyes, X. Zhao, D.L. Neuhoff, and T.N. Pappas, "Lossy compression of bilevel images based on Markov random fields," in *IEEE ICIP*, 2007, vol. 2, pp. II–373.

[7] M.G. Reyes and D.L. Neuhoff, "Arithmetic encoding of Markov random fields," in *IEEE ISIT*, 2009, pp. 532–536.

[8] M.G. Reyes and D.L. Neuhoff, "Lossless reduced cutset coding of Markov random fields," in *IEEE Data Compression Conference (DCC)*, 2010, pp. 386–395.

[9] A. Farmer, A. Josan, M.A. Prelee, D.L. Neuhoff, and T.N. Pappas, "Cutset sampling and reconstruction of images," in *IEEE ICIP*, 2011, pp. 1909–1912.

[10] M.A. Prelee, D.L. Neuhoff, and T.N. Pappas, "Image reconstruction from a manhattan grid via piecewise plane fitting and gaussian Markov random fields," in *IEEE ICIP*, 2012, pp. 2061–2064.

[11] M.A. Prelee and D.L. Neuhoff, "A sampling theorem for manhattan grids," in *IEEE ICASSP*, 2012, pp. 3797–3800.

[12] M.G. Rabbat and R.D. Nowak, "Quantized incremental algorithms for distributed optimization," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 798–808, 2005.

[13] D. Li and Y.H. Hu, "Energy-based collaborative source localization using acoustic microsensor array," *EURASIP Journal on Advances in Signal Processing*, vol. 2003, no. 4, pp. 321–337, 2003.