

# ROBUST DECENTRALIZED SOURCE LOCALIZATION VIA AVERAGING

*Michael Rabbat, Robert Nowak, and James Bucklew*

Department of Electrical and Computer Engineering, University of Wisconsin-Madison

1415 Engineering Drive, Madison, Wisconsin, 53706

Email: rabbat@cae.wisc.edu, {nowak, bucklew}@engr.wisc.edu

## ABSTRACT

We present a new approach to localizing an isotropic energy source using measurements from distributed sensors based on kernel averaging techniques. The location estimate is easily and efficiently calculated in a decentralized fashion. Statistical properties are derived for a very general measurement model. Experiments suggest that the proposed estimator is much more robust and exhibits better performance characteristics than the popular least squares estimator under a variety of conditions.

## 1. LOCALIZATION VIA AVERAGING

The problem of localizing and tracking an energy-emitting source encompasses many of the challenging issues which commonly arise in wireless sensor network applications [1]. Consequently, this problem has recently received a great deal of attention. In [2], Chen et al. present an approach to source localization using direction of arrival (DOA) measurements, with a lot of practical insight. Sheng and Hu consider the localization problem using received signal strength (RSS) measurements corrupted with zero mean Gaussian noise in [3]. The maximum likelihood estimate in this context amounts to solving a nonlinear least squares optimization problem. They formulate a variety of approaches to solving this problem, most of which are not easily implemented in a decentralized fashion and are too computationally complex even for centralized computation with measurements from more than a handful of nodes. In our previous work, [4], we propose and analyze an energy-efficient decentralized method for approximating the nonlinear least squares solution. Wang et al. have compared bounds on the best localization performance achievable using time of arrival (TOA), RSS, and DOA measurements under a Gaussian noise model [5]. Their results indicate that TOA and RSS measurements offer comparable performance, both of which are superior to that achievable using DOA measurements.

This paper introduces a new approach to localizing an isotropic energy source using measurements from distributed sensors. Our aim is to design a practical algorithm which is implementable in a decentralized fashion and which is robust to unknown source and propagation parameters, as well as noise. Let  $\theta$  denote the unknown source location, and let  $x_j$ ,  $j = 1, \dots, n$  denote the known locations of  $n$  sensor nodes. Throughout this paper we focus on the case where  $\theta$  and  $x_j$  are two dimensional but our results carry over to the three dimensional case. We assume that sensors are deployed uniformly at random over the unit square. Each sensor takes measurements according to the model

$$y_j = \alpha \rho(\|x_j - \theta\|) + w_j, \quad (1)$$

where  $\alpha > 0$  is a signal strength,  $\rho : \mathbb{R}^+ \rightarrow \mathbb{R}^+$  is a monotone decreasing function, and  $w_j$  are statistically independent, bounded, non-negative random variables with distribution  $p(w)$ . Throughout this paper  $\|\cdot\|$  denotes Euclidean distance. Although this model is very general in that it encompasses both RSS and TOA, our main focus is on RSS measurements where the decay function has the usual form

$$\rho(\|x_j - \theta\|) = \frac{1}{\|x_j - \theta\|^\beta}, \quad (2)$$

and typical values of  $\beta$  are between 2 and 4. See [6] for a discussion of the merits of both types of measurements in localization problems. This model is physically motivated by the common perception of sensor network nodes as being cheap, wireless devices that probably implement a very simple procedure for collecting measurements. In particular, we envision RSS sensors composed of a square-law concatenated with an integrator. Thus, all measurements will be nonnegative. Furthermore, the range of measurements is bounded due to basic physical considerations.

To avoid boundary effects near the edge of the sensor field (i.e., unit square), throughout this paper we assume that the source is located well within the interior of the region being sensed. There are a number of ways of formalizing this notion which will arise in what follows. See [5] for a discussion of the coverage areas for different sensing modalities.

Our approach to source localization is based on kernel averaging estimators which are simply illustrated as follows. Consider, for the time being, an idealized situation where there is no noise and each sensor is able to determine whether it is within a distance of  $\gamma$  from the source. The network then estimates the source location by averaging the locations of those sensors that are “close enough” to the source. Using the indicator function  $1_{\{\cdot\}}$ , we can write this estimator as

$$\hat{\theta}_1 = \frac{\sum_{j=1}^n x_j 1_{\{\|x_j - \theta\| \leq \gamma\}}}{\sum_{j=1}^n 1_{\{\|x_j - \theta\| \leq \gamma\}}}. \quad (3)$$

It is easy to show that such an estimator is unbiased and consistent, but there are many issues which need to be addressed to go from this simple concept to a practical estimation scheme. Sensors do not directly know their distance from the source. Rather, the measurements are generally some complicated nonlinear function of the sensor's distance to the source. For example, if the measurements obey (2) then  $\|x_j - \theta\| \approx (\alpha/y_j)^{1/\beta}$ . The parameters  $\alpha$  and  $\beta$  are typically not known exactly. Additionally, it is reasonable to suppose that sensors located closer to the source have more reliable measurements than those who are far away from the source [7, 8]. We would probably like to modify the estimator (3) to account for

this by giving sensors that are further from the source less influence on the estimate. Accordingly, we propose estimators of the general form

$$\hat{\theta} = \frac{\sum_{j=1}^n x_j g(y_j)}{\sum_{j=1}^n g(y_j)}, \quad (4)$$

where  $g : \mathbb{R}^+ \rightarrow \mathbb{R}^+$  is a monotone increasing function with  $g(0) = 0$  and  $\lim_{y \rightarrow \infty} g(y) < \infty$ . Thus, we have  $g$  filling two roles. First it is “inverting” the decay function  $\rho$  in a very loose sense to recover the distance from each sensor to the source. Simultaneously,  $g$  dictates how greatly a contribution different sensors make in the computation of (4) based on their received signal strength (estimated distance from the source).

## 2. DECENTRALIZED ALGORITHMS FOR COMPUTATION

Regarding the decentralized computation of such an estimator, first observe that  $\hat{\theta}$  can be written as the quotient of two averages:  $\frac{1}{n} \sum_{j=1}^n x_j g(y_j)$  and  $\frac{1}{n} \sum_{j=1}^n g(y_j)$ . The problem of computing averages in a decentralized manner has been studied in a number of contexts and is an instance of the so-called Consensus Problem. A number of algorithms have been proposed including those discussed in [9, 10, 11, 12]. All of these algorithms only involve communications between neighboring nodes in the sensor network communication graph. That is, information is only exchanged between nodes that communicate directly.

One class of algorithms for computing averages falls under the general heading of *local broadcast algorithms*, so named because at each iteration each node broadcasts its local value to each of its immediate neighbors. After learning the values of its neighbors, each node updates its local value according to a weighted average of the neighbors’ values and its previous local value. Then the nodes broadcast the updated values and repeat. The evolution of local estimated values can be modelled as a Markov Chain, where the weights used by each node in computing the weighted average update at each iteration correspond to the columns of the probability transition matrix. It is easy to verify that when the corresponding stationary distribution is uniform over all nodes in the network then the values at each iteration converge to the average of the initial values at each node.

So-called *token passing algorithms* offer an alternative approach to computing averages in a decentralized fashion. In this approach, the node in possession of the token at each iteration broadcasts its current value to all neighboring nodes who then update their local values accordingly. The token takes a random walk through the network corresponding to a random walk on a graph which also corresponds to a Markov chain. As with the local broadcast approach, if the stationary distribution of the Markov chain is uniform over all nodes then the value computed at each node converges to the average almost surely.

There are a number of heuristics for determining appropriate weights or transition probabilities for both of these approaches which only require each node in the network to know information about the connectivity of its neighbors, including the popular Metropolis-Hastings method. Thus, these algorithms show great promise for robust computation in completely decentralized networked systems. Both approaches are robust in the sense that eventually all sensors know the average and thus the estimated value.

If one node (or a subset of nodes) are compromised then the estimated source location is not lost.

A local broadcast algorithm usually converges much faster than token passing algorithms because more information is exchanged at each iteration, however the local broadcast algorithm requires more communication at each iteration and thus may be less desirable from the perspective of energy expended. Determining the optimal approach (e.g., in terms of latency, accuracy, or energy) is an interesting and relevant question to the sensor network community. This issue is not addressed within this paper due to space limitations, however it is the subject of our ongoing research. We merely wish to point out at this time that there are many efficient decentralized schemes for computing averages in networked systems.

In the following sections we analyze some asymptotic statistical properties of the kernel estimator. Experiments reveal that our proposed averaging estimator performs better than the popular maximum likelihood or least squares estimator in a number of scenarios and is much more robust to modelling errors.

## 3. BASIC PROPERTIES

### 3.1. The Case of No Noise

The following theorem characterizes the performance of the kernel averaging estimator when there is no noise. The only source of variability in this case is the randomness in the sensor locations. We define the variance of an unbiased position estimator to be  $\text{Var}(\hat{\theta}) = \mathbb{E}[(\hat{\theta}(1) - \theta(1))^2] + \mathbb{E}[(\hat{\theta}(2) - \theta(2))^2]$ , the trace of the covariance matrix of  $\hat{\theta}$ . As stated before, we assume that  $g : \mathbb{R}^+ \rightarrow \mathbb{R}^+$  is a monotone increasing function with  $\lim_{y \rightarrow \infty} g(y) < \infty$ . Moreover, we assume that the support of  $g$  to be over the interval  $[y_{\min}, \infty]$  for some  $a > 0$  and that  $\int_0^\infty r g(\alpha \rho(r)) dr = 1/(2\pi)$ . Thus, for a sensor to contribute to the computation (4) it must measure a value  $y > y_{\min}$ , or, equivalently, it must be within a distance of  $R$  of the source location where  $R = \rho^{-1}(y_{\min}/\alpha)$ . Additionally we assume that the true source location is far enough away from the field border so that nodes at the edge of the field only ever measure noise. These assumptions ensure that there are no boundary artifacts.

**Theorem 1** *The estimator  $\hat{\theta}$  given by (4) is unbiased. Furthermore, the asymptotic variance constant for this estimator is given by*

$$\lim_{n \rightarrow \infty} n \text{Var}(\hat{\theta}) = 2\pi \int_0^R r^3 g^2(\alpha \rho(r)) dr.$$

**Remark:** The asymptotic variance constant provides a very useful characterization of estimator performance. For example, we can use asymptotic variance constants to compare various estimators or, if knowledge of the source and propagation characteristics are known, then we can design kernels which minimize the asymptotic variance constant.

*Sketch of proof:* We first verify that the estimator is unbiased. Because  $g$  only has support on the interval  $[y_{\min}, \infty]$ , only those sensors that are within  $R$  of the true source location contribute non-zero values  $g(y_j)$ . Moreover, because the true source location is sufficiently away from the boundary of the sensed region there is a circle of radius  $R$  centered at the true source location which lies completely within the unit square. It follows that the conditional distribution of nodes within this circle is uniform since the entire

circle is within the unit square. Thus, by conditioning on the set  $Y = \{y_j, j = 1, \dots, n\}$ , one establishes that  $\mathbb{E}[\hat{\theta}] = \theta$ .

Observe that the averaging estimator automatically adapts to the signal strength parameter. One can show that as  $n \rightarrow \infty$ , under the conditions on  $g$  listed in the previous section,  $\frac{1}{n} \sum_{j=1}^n x_j g(y_j) \rightarrow \theta \alpha^2$  and  $\frac{1}{n} \sum_{j=1}^n g(y_j) \rightarrow \alpha^2$ . Thus, one can disregard  $\alpha$  when designing the function  $g$ , or, in other words, the kernel averaging estimator is invariant to  $\alpha$ .

The variance calculation is the sum of two components of the form

$$\mathbb{E} \left[ \left( \frac{\sum_{j=1}^n (x_j(i) - \theta(i)) g(y_j)}{\sum_{j=1}^n g(y_j)} \right)^2 \right],$$

for  $i = 1, 2$ . Note that, for  $j \neq k$  we have

$$\mathbb{E}[(x_j(i) - \theta(i))(x_k(i) - \theta(i)) | y_j, y_k] = 0.$$

Thus, by conditioning on  $Y$  again, the cross terms in (5) cancel and we can write

$$\begin{aligned} \text{Var}(\hat{\theta}) &= \mathbb{E} \left[ \frac{\sum_{j=1}^n \|x_j - \theta\|^2 g^2(y_j)}{\left( \sum_{j=1}^n g(y_j) \right)^2} \right] \\ &= \frac{1}{n} \mathbb{E} \left[ \frac{\frac{1}{n} \sum_{j=1}^n \|x_j - \theta\|^2 g^2(y_j)}{\left( \frac{1}{n} \sum_{j=1}^n g(y_j) \right)^2} \right]. \end{aligned}$$

Both the numerator and denominator of the term inside the expectation form a sequence of bounded random variables which converge almost surely and thus the expectations will converge. Thus,

$$\begin{aligned} \lim_{n \rightarrow \infty} n \text{Var}(\hat{\theta}) &= \frac{\mathbb{E}[\|x - \theta\|^2 g^2(\alpha \rho(\|x - \theta\|))]}{(\mathbb{E}[g(\alpha \rho(\|x - \theta\|))])^2} \\ &= \frac{2\pi \int_0^{\alpha \rho(R)} r^3 g^2(\alpha \rho(r)) dr}{\left( 2\pi \int_0^{\alpha \rho(R)} r g(\alpha \rho(r)) dr \right)^2}. \end{aligned}$$

Further algebraic manipulations yield the desired result.

### 3.2. The Case with Noise

In the more general case where the measurements are corrupted with noise. Under a variety of assumptions that allow us to avoid bias due to “edge effects” (e.g., assuming that the noise is bounded and placing further restrictions on  $\theta$ , or simply considering the case where  $\theta$  is at the origin for asymptotic purposes), by the same arguments given in the previous section the estimator  $\hat{\theta}$  is unbiased and the asymptotic variance constant is given by

$$\begin{aligned} \lim_{n \rightarrow \infty} n \text{Var}(\hat{\theta}) &= \frac{\mathbb{E}[\|x - \theta\|^2 g^2(y_j)]}{(\mathbb{E}[g(y_j)])^2} \\ &= \frac{2\pi \int_0^\infty \int_0^\infty r^3 g^2(\alpha \rho(r) + w) p(w) dw dr}{\left( 2\pi \int_0^\infty \int_0^\infty r g(\alpha \rho(r) + w) p(w) dw dr \right)^2}, \end{aligned}$$

where  $p(w)$  is the noise distribution.

## 4. LEAST SQUARES ESTIMATION

For the sake of comparison, we briefly discuss statistical properties of the least squares (LS) estimate which corresponds to the maximum likelihood estimate under a Gaussian noise model. Suppose our observation model is as in (1) and let  $\bar{w}$  denote the mean of the noise variables  $w_j$ . Assuming that  $\alpha$ ,  $\beta$ , and  $\bar{w}$  are known, the least squares estimator is defined as

$$\hat{t}_{LS} = \arg \min_t \frac{1}{2} \sum_{j=1}^n \left( \frac{\alpha}{\|x_j - t\|^\beta} + \bar{w} - y_j \right)^2. \quad (5)$$

The LS estimate is generally held as the “gold standard” for unbiased parametric estimators. It can be viewed as a parametric estimator for  $\theta$  which minimizes the mean squared error between the observed measurement values and the values predicted by the model, and it can be effective when the model is accurate. However, a Gaussian noise model is often adopted despite whether it may or may not be the most accurate noise model. Because of this and because it has the appealing interpretation as finding the best match to a model the LS estimator is commonly adopted.

A major drawback of LS estimators in general is that they are highly sensitive to mismodelling. The solution to the minimization problem (5) is a nonlinear function of the  $y_j$  and if the model parameters are not known precisely then the LS estimate will become biased and generally exhibits poor performance. In contrast, the proposed kernel method is essentially linear (linear plus a simple normalization). Consequently, the kernel averaging estimator is also very robust to quantization errors. The LS estimator, on the other hand, is much more sensitive to quantization, and the effects of quantization are much more complicated to analyze [10]. In the following sections we illustrate via experiments that the proposed kernel averaging estimator is much more robust to modelling errors. Additionally, our analysis of the asymptotic variance constants of these estimators (not presented in this paper due to space limitations) indicates that the kernel averaging estimator outperforms the LS estimator in many reasonable scenarios.

As a result its complicated nature, decentralized computation of the LS estimator is much more problematic than that of our proposed kernel approach. In [4] we introduced a decentralized scheme for computing (5). Unlike the distributed algorithms discussed in Section 2, this scheme requires the actual estimate to be passed through the network from node to node as opposed to the exchange of local values. Such an approach to decentralized computation may easily be compromised if communication between nodes in the network is unreliable. If the actual estimate is lost or corrupted during transmission then, essentially, the computation must be restarted from scratch. In comparison, because the distributed averaging schemes discussed in Section 2 only involve passing of local information (with the actual estimate computed independently at each node), if one transmission is corrupted or if a node goes down the overall network-wide computation is not affected. Moreover, the LS estimate is actually much more complicated to compute in general (*even in a centralized scenario!*) because the problem (5) is in general a highly nonconvex optimization problem with many local minima. Thus the kernel averaging estimator is attractive from a number of perspectives.

## 5. SIMULATION RESULTS

In this section we compare the performance of the kernel averaging estimator with the least squares estimator via simulated ex-

true $\beta$	u = 10		u = 50		u = 100	
	AV	LS	AV	LS	AV	LS
1.75	0.023105	0.032908	0.029043	0.184190	0.032153	0.246440
2.0	0.018584	0.012366	0.025847	0.095909	0.028160	0.153190
2.25	0.019456	0.021119	0.025604	0.049594	0.026357	0.119890
3.0	0.021134	0.038980	0.023623	0.058753	0.021154	0.064065

**Table 1.** Average MSE for the Kernel Averaging (AV) and Least Squares Estimators over a network of 100 nodes. Each value is the average performance over 100 trials.

periments. Nodes are deployed uniformly at random over the unit square and a source is located at random within the region  $[0.4, 0.6] \times [0.4, 0.6]$ . The nodes make measurements according to (1) with  $\rho$  as in (2). The noise variables  $w_j$  are distributed uniformly over the interval  $[0, u]$ , where  $u$  is varied in our experiments. This noise model could represent uncertainty in the measurements, in propagation of the energy signal, in the system hardware, and in a number of other facets of the problem. In all of our simulations we use  $\alpha = 1$ . The LS estimator is given the correct value of  $\alpha$  and is designed assuming  $\beta = 2$ . The kernel  $g$  for the averaging estimator is optimized (with respect to the asymptotic variance constant) for the case where  $\beta = 2$ . Recall that the kernel averaging estimator automatically adapts to the signal strength (i.e., it is invariant to  $\alpha$ ), thus it does not need to be given the true value.

We compare the performance and robustness of both estimators to modelling errors and varying noise levels. Table 1 shows the average mean squared error under a number of different settings for  $u$  and  $\beta$ , for simulated networks of 100 nodes. Each value in the table corresponds to the mean squared error for both estimators, averaged over 100 trials. Clearly, the kernel averaging estimator performs as well, and often much better in nearly every situation. Observe that the second row, where  $\beta = 2$  corresponds to the case where there is no mismodelling and in this case the LS estimator outperforms the kernel averaging estimator at 2 out of 3 noise levels. Additionally, note that the kernel averaging estimator exhibits a graceful degradation behavior across the board whereas the performance of the LS estimator decays much at higher noise levels or modelling errors.

## 6. DISCUSSION

The experimental results of the previous section illustrate that the kernel averaging approach to source localization performs well and is robust to modelling errors. The simple form (normalized average) of the kernel method makes it much more robust to mismodelling and noise than a least squares estimator. Simulations also indicate that the averaging estimator performs well even when the decay parameter,  $\beta$ , is not known exactly. It may be possible to modify both the least squares and kernel averaging estimators so that they jointly estimate (or are adaptive to)  $\beta$ , but this will add to the nonlinearity of either scheme and could worsen performance. However, this is an open issue which we plan to investigate further.

Another benefit of the averaging estimator over the least squares estimator is that it is much simpler to compute. The optimal least squares location estimator is generally a nonlinear function of the data and requires a complicated nonconvex optimization. We also remark that the least squares estimator described in Section 4 is specifically designed for an isotropic energy source. If the source is not isotropic then a different model must be formu-

lated and direction or orientation parameters must also be specified or estimated. On the other hand, the kernel averaging estimator simply requires that the mean of the energy distribution be identical to the true source location. For example, if the energy profile is ellipsoidal or otherwise symmetric about the source location, this will still be acceptable for the kernel averaging method. Ultimately we would like to develop a practical, robust, decentralized scheme for detecting the presence of, determining the number of, and localizing multiple sources. This is a very challenging problem and there are many open issues which still need to be addressed.

## 7. REFERENCES

- [1] F. Zhao and L. Guibas, *Wireless Sensor Networks: An Information Processing Approach*, Morgan Kaufmann, 2004.
- [2] J.C. Chen, K. Yao, and R.E. Hudson, "Source localization and beamforming," *IEEE Sig. Proc. Magazine*, March 2002.
- [3] X. Sheng and Y.H. Hu, "Energy based acoust source localization," in *Proc. IPSN*, Palo Alto, CA, April 2003.
- [4] M.G. Rabbat and R.D. Nowak, "Decentralized source localization and tracking," in *Proc. IEEE ICASSP*, Montreal, Canada, May 2004.
- [5] H. Wang, L. Yip, K. Yao, and D. Estrin, "Lower bounds of localization uncertainty in sensor networks," in *Proc. IEEE ICASSP*, Montreal, Canada, May 2004.
- [6] N. Patwari, A.O. Hero, M. Perkins, N.S. Correal, and R.J. O'Dea, "Relative location estimation in sensor networks," *IEEE Trans. on Signal Processing*, vol. 51, no. 9, pp. 2219–2228, Aug. 2003.
- [7] H. Wang, K. Yao, G. Pottie, and D. Estrin, "Entropy-based sensor selection heuristic for localization," in *Proc. IPSN*, Berkeley, California, April 2004.
- [8] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration for tracking applications," *IEEE Sig. Proc. Mag.*, March 2002.
- [9] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems and Control Letters*, vol. 53, pp. 65–78, 2004.
- [10] M.G. Rabbat and R.D. Nowak, "Quantized incremental algorithms for distributed optimization," To appear in *IEEE Journal on Selected Areas in Communications*, 2005.
- [11] R. Olfati-Saber and R.M. Murray, "Consensus protocols for networks of dynamic agents," in *Proc. American Control Conference*, Denver, Colordao, June 2003.
- [12] M. Alanyali, S. Venkatesh, O. Savas, and S. Aeron, "Distributed bayesian hypothesis testing in sensor networks," in *Proc. Amer. Control Conf.*, Boston, Mass., June 2004.