

Research Article

Distributed Encoding Algorithm for Source Localization in Sensor Networks

Yoon Hak Kim¹ and Antonio Ortega²

¹System LSI Division, Samsung Electronics, Giheung campus, Gyeonggi-Do 446-711, Republic of Korea

²Department of Electrical Engineering, Signal and Image Processing Institute, University of Southern California, Los Angeles, CA 90089-2564, USA

Correspondence should be addressed to Yoon Hak Kim, yhk418@gmail.com

Received 12 May 2010; Accepted 21 September 2010

Academic Editor: Erchin Serpedin

Copyright © 2010 Y. H. Kim and A. Ortega. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We consider sensor-based distributed source localization applications, where sensors transmit quantized data to a fusion node, which then produces an estimate of the source location. For this application, the goal is to minimize the amount of information that the sensor nodes have to exchange in order to attain a certain source localization accuracy. We propose a distributed encoding algorithm that is applied after quantization and achieves significant rate savings by merging quantization bins. The bin-merging technique exploits the fact that certain combinations of quantization bins at each node cannot occur because the corresponding spatial regions have an empty intersection. We apply the algorithm to a system where an acoustic amplitude sensor model is employed at each node for source localization. Our experiments demonstrate significant rate savings (e.g., over 30%, 5 nodes, and 4 bits per node) when our novel bin-merging algorithms are used.

1. Introduction

In sensor networks, multiple correlated sensor readings are available from many sensors that can sense, compute and communicate. Often these sensors are battery-powered and operate under strict limitations on wireless communication bandwidth. This motivates the use of data compression in the context of various tasks such as detection, classification, localization, and tracking, which require data exchange between sensors. The basic strategy for reducing the overall energy usage in the sensor network would then be to decrease the communication cost at the expense of additional computation in the sensors [1].

One important sensor collaboration task with broad applications is source localization. The goal is to estimate the location of a source within a sensor field, where a set of distributed sensors measures acoustic or seismic signals emitted by a source and manipulates the measurements to produce meaningful information such as signal energy, direction-of-arrival (DOA), and time difference-of-arrival (TDOA) [2, 3].

Localization based on acoustic signal energy measured at individual acoustic amplitude sensors is proposed in [4], where each sensor transmits unquantized acoustic energy readings to a fusion node, which then computes an estimate of the location of the source of these acoustic signals. Localization can be also performed using DOA sensors (sensor arrays) [5]. The sensor arrays generally provide better localization accuracy, especially in far field, as compared to amplitude sensors, while they are computationally more expensive. TDOA can be estimated by using various correlation operations and a least squares (LS) formulation can be used to estimate source location [6]. Good localization accuracy for the TDOA method can be accomplished if there is accurate synchronization among sensors, which will tend to require a high cost in wireless sensor networks [3].

None of these approaches take explicitly into account the effect of sensor reading quantization. Since practical systems will require quantization of sensor readings before transmission, estimation algorithms will be run on quantized sensor readings. Thus, it would be desirable to minimize the information in terms of rate before being transmitted

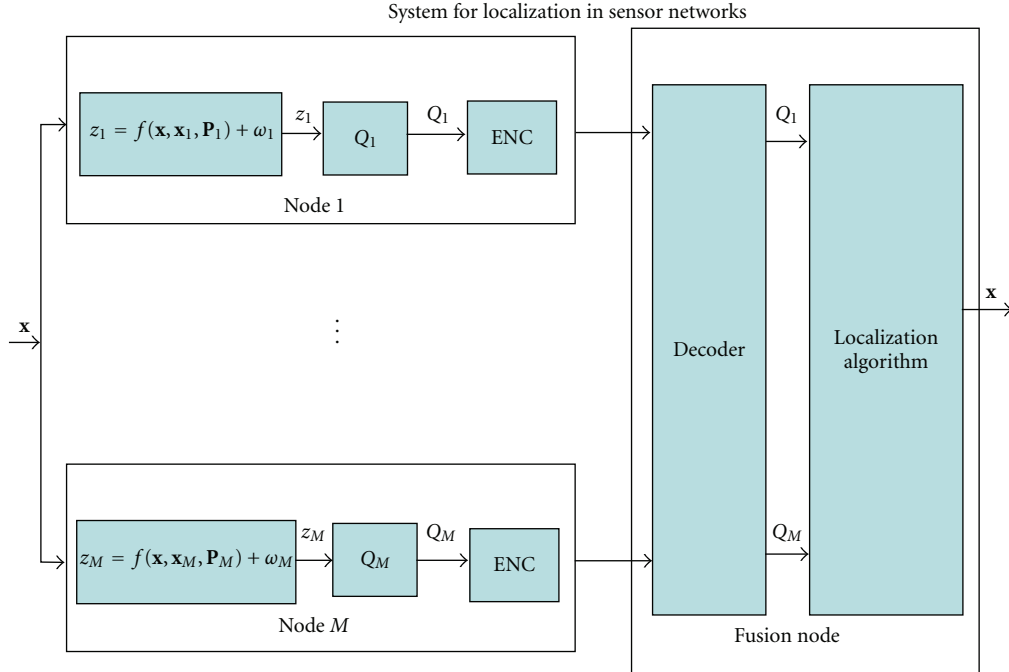


FIGURE 1: Block diagram of source localization system. We assume that the channel between each node and fusion node is noiseless and each node sends its quantized (Quantizer, Q_i) and encoded (ENC block) measurement to the fusion node, where decoding and localization are conducted in a distributed manner.

to a fusion node. It is noted that there exists some degree of redundancy between the quantized sensor readings since each sensor collects information (e.g., signal energy or direction) regarding a source location. Clearly, this redundancy can be reduced by adopting distributed quantizers designed to maximize the localization accuracy by exploiting the correlation between the sensor readings (see [7, 8]).

In this paper, we observe that the redundancy can be also reduced by encoding the quantized sensor readings for a situation, where a set of nodes (Each node may employ one sensor or an array of sensors, depending on the applications) and a fusion node wish to cooperate to estimate a source location (see Figure 1). We assume that each node can estimate noise-corrupted source characteristics (z_i in Figure 1), such as signal energy or DOA, using actual measurements (e.g., time-series measurements or spatial measurements). We also assume that there is only one way communication from nodes to the fusion node; that is, there is no feedback channel, the nodes do not communicate with each other (no relay between nodes), and these various communication links are reliable.

In our problem, a source signal is measured and quantized by a series of distributed nodes. Clearly, in order to make localization possible, each possible location of the source produces a different vector of sensor readings at the nodes. Thus, the vector of the readings (z_1, \dots, z_M) should uniquely define the localization. Quantization of the readings at each node reduces the accuracy of the localization. Each quantized value (e.g., Q_i at node i) of a sensor reading can then be linked to a region in space, where the source can be found. For example, if distance information is provided by

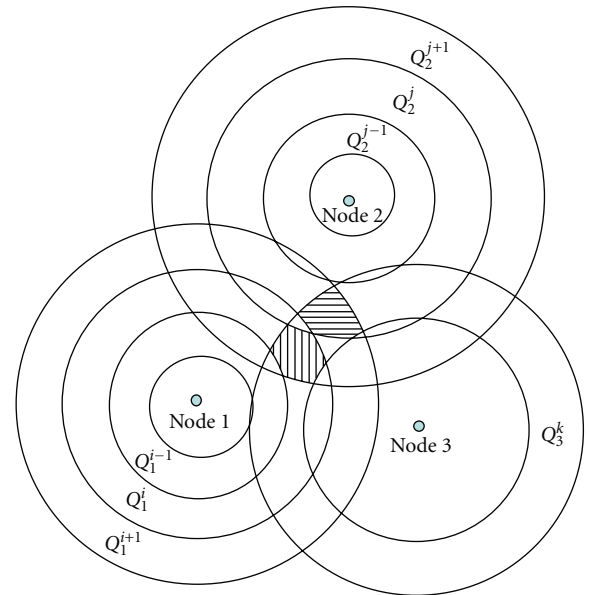


FIGURE 2: Simple example of source localization, where an acoustic amplitude sensor is employed at each node. The shaded regions refer to nonempty intersections, where the source can be found.

sensor readings, the regions corresponding to sensor readings will be circles centered on the nodes and thus quantized values of those readings will then be mapped to “rings” centered on the nodes. Figure 2 illustrates the case, where 3 nodes equipped with acoustic amplitude sensors measure

the distance information for source localization. Denote Q_i^j the j th quantization bin at node i ; that is, whenever sensor reading z_i at node i belongs to j th bin, the node will transmit Q_i^j to the fusion node. From the discussion, it should be clear that since each quantized sensor reading Q_i can be associated with the corresponding ring, the fusion node can locate the source by computing the intersection of those 3 rings from the combination (Q_1, Q_2, Q_3) received from the 3 nodes. (In a noiseless case, there always exists a nonempty intersection corresponding to each received combination, where a source is located. However, empty intersections may be constructed in a noisy case. In Figure 2, suppose that node 2 transmits Q_2^{j-1} instead of Q_2^j due to measurement noise. Then, the fusion node will receive (Q_1, Q_2^{j-1}, Q_3^k) which leads to an empty intersection. Probabilistic localization methods should be employed to handle empty intersections. For further details, see [9].) Therefore, the combinations such as $(Q_1^i, Q_2^{j+1}, Q_3^k)$ or (Q_1^i, Q_2^j, Q_3^k) transmitted from the nodes will tend to produce nonempty intersections (the shaded regions in Figure 2, resp.) while numerous other combinations randomly collected may lead to empty intersections, implying that such combinations are very unlikely to be transmitted from the nodes (e.g., $(Q_1^{i+1}, Q_2^{j-1}, Q_3^k)$, $(Q_1^{i-1}, Q_2^{j-1}, Q_3^k)$, and many others). In this work, we focus on developing tools that allow us to exploit this observation in order to eliminate the redundancy. More specifically, we consider a novel way of reducing the effective number of quantization bins consumed by all the nodes involved while preserving localization performance. Suppose that one of the nodes reduces the number of bins that are being used. This will cause a corresponding increase of uncertainty. However, the fusion node that receives a combination of the bins from all the nodes should be able to compensate for the increase by using the data from the other nodes as side information.

We propose a novel distributed encoding algorithm that allows us to achieve significant rate savings [8, 10]. With our method, we merge (non-adjacent) quantization bins in a given node whenever we determine that the ambiguity created by this merging can be resolved at the fusion node once information from other nodes is taken into account. In [11], the authors focused on encoding the correlated measurements by merging the adjacent quantization bins at each node so as to achieve rate savings at the expense of distortion. Notice that they search the quantization bins to be merged that show redundancy in encoding perspective while we find the bins for merging that produce redundancy in localization perspective. In addition, while in their approach each computation of distortion for pairs of bins will be required to find the bins for merging, we develop simple techniques that choose the bins to be merged in a systematic way.

It is noted that our algorithm is an example of binning as can be found in Slepian-Wolf and Wyner-Ziv techniques [11, 12]. In our approach, however, we achieve rate savings purely through binning and provide several methods to select candidate bins for merging. We apply our distributed encoding algorithm to a system, where an acoustic amplitude

sensor model proposed in [4] is considered. Our experiments show rate savings (e.g., over 30%, 5 nodes, and 4 bits per node) when our novel bin-merging algorithms are used.

This paper is organized as follows. The terminologies and definitions are given in Section 2, and the motivation is explained in Section 3. In Section 4, we consider quantization schemes that can be used with the encoding at each node. An iterative encoding algorithm is proposed in Section 5. For a noisy situation, we consider the modified encoding algorithm in Section 6 and describe the decoding process and how to handle decoding errors in Section 7. In Section 8, we apply our encoding algorithm to the source localization system, where an acoustic amplitude sensor model is employed. Simulation results are given in Section 9, and the conclusions are found in Section 10.

2. Terminologies and Definitions

Within the sensor field S of interest, assume that there are M nodes located at known spatial locations, denoted $\mathbf{x}_i, i = 1, \dots, M$, where $\mathbf{x}_i \in S \subset \mathbf{R}^2$. The nodes measure signals generated by a source located at an unknown location $\mathbf{x} \in S$. Denote by z_i the measurement (equivalently, sensor reading) at the i th node over a time interval k

$$z_i(\mathbf{x}, k) = f(\mathbf{x}, \mathbf{x}_i, \mathbf{P}_i) + w_i(k) \quad \forall i = 1, \dots, M, \quad (1)$$

where $f(\mathbf{x}, \mathbf{x}_i, \mathbf{P}_i)$ denotes the sensor model employed at node i and the measurement noise $w_i(k)$ can be approximated using a normal distribution, $N(0, \sigma_i^2)$. (The sensor models for acoustic amplitude sensors and DOA sensors can be expressed in this form [4, 13].) \mathbf{P}_i is the parameter vector for the sensor model (an example of \mathbf{P}_i for an acoustic amplitude sensor case is given in Section 8). It is assumed that each node measures its sensor reading $z_i(\mathbf{x}, k)$ at time interval k , quantizes it and sends it to a fusion node, where all sensor readings are used to obtain an estimate $\hat{\mathbf{x}}$ of the source location.

At node i , we use a R_i -bit quantizer with a dynamic range $[z_{i,\min} \ z_{i,\max}]$. We assume that the quantization range can be selected for each node based on desirable properties of their respective sensing ranges [14]. Denote by $\alpha_i(\cdot)$ the quantizer with quantization level L_i at node i , which generates a quantization index $Q_i \in I_i = \{1, \dots, 2^{R_i} = L_i\}$. In what follows, Q_i will be also used to denote the quantization bin to which measurement z_i belongs.

This formulation is general and captures many scenarios of practical interest. For example, $z_i(\mathbf{x}, k)$ could be the energy captured by an acoustic amplitude sensor (this will be the case study presented in Section 8), but it could also be a DOA measurement. (In the DOA case, each measurement at a given node location will be provided by an array of collocated sensors.) Each scenario will obviously lead to a different sensor model $f(\mathbf{x}, \mathbf{x}_i, \mathbf{P}_i)$. We assume that the fusion node needs measurements, $z_i(\mathbf{x}, k)$, from *all* nodes in order to estimate the source location.

Let $S_M = I_1 \times I_2 \times \dots \times I_M$ be the cartesian product of the sets of quantization indices. S_M contains $|S_M| = (\prod_i^M L_i)$ M -tuples representing all possible combinations of quantization indices

$$S_M = \{(Q_1, \dots, Q_M) \mid Q_i = 1, \dots, L_i, i = 1, \dots, M\}. \quad (2)$$

We denote S_Q the subset of S_M that contains all the quantization index combinations that can occur in a real system, that is, all those generated as a source moves around the sensor field and produces readings at each node

$$S_Q = \{(Q_1, \dots, Q_M) \mid \exists \mathbf{x} \in S, Q_i = \alpha_i(z_i(\mathbf{x})), i = 1, \dots, M\}. \quad (3)$$

For example, assuming that each node measures noiseless sensor readings (i.e., $w_i = 0$), we can construct the set S_Q by collecting only the combinations that lead to nonempty intersections. (The combinations $(Q_1^i, Q_2^{j+1}, Q_3^k)$, (Q_1^i, Q_2^j, Q_3^k) corresponding to the shaded regions in Figure 2 will belong to S_Q .) In a noisy situation, how to construct S_Q will be further explained in Section 6.

We denote S_i^j the subset of S_Q that contains all M -tuples in which the i th node is assigned the j th quantization bin

$$S_i^j = \{(Q_1, \dots, Q_M) \in S_Q \mid Q_i = j\}, \quad (4)$$

$$i = 1, \dots, M, j = 1, \dots, L_i.$$

This set will provide all possible combinations of $(M - 1)$ tuples that can be transmitted from other nodes when the j th bin at node i was actually transmitted. In other words, the fusion node will be able to identify which bin actually occurred at node i by exploiting the set as side information, when there is uncertainty induced by merging bins at node i .

Since $(M - 1)$ quantized measurements out of each M -tuple in S_i^j are used in actual process of encoding, it would be useful to construct the set of $(M - 1)$ tuples generated from S_i^j . We denote by \overline{S}_i^j the set of $(M - 1)$ -tuples obtained from M -tuples in S_i^j , where only the quantization bins at positions other than position i are stored. That is, if $\mathbf{Q} = (Q_1, \dots, Q_M) = (a_1, \dots, a_M) \in S_i^j$, then we always have $(a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_M) \in \overline{S}_i^j$. Clearly, there is one to one correspondence between the elements in S_i^j and \overline{S}_i^j , so that $|S_i^j| = |\overline{S}_i^j|$.

3. Motivation: Identifiability

In this section, we assume that $\Pr[(Q_1, \dots, Q_M) \in S_Q] = 1$; that is, only combinations of quantization indices belonging to S_Q can occur and those combinations belonging to $S_M - S_Q$ never occur. These sets can be easily obtained when there is no measurement noise (i.e., $w_i = 0$) and no parameter mismatches. As discussed in the introduction, there will be numerous elements in S_M that are not in S_Q . Therefore, simple scalar quantization at each node would be inefficient because a standard scalar quantizer would allow

us to represent any of the M -tuples in S_M . What we would like to determine now is a method such that independent quantization can still be performed at each node, while at the same time, we reduce the redundancy inherent in allowing all the combinations in S_M to be chosen. Note that, in general, determining that a specific quantizer assignment in S_M does not belong to S_Q requires having access to the whole vector, which obviously is not possible if quantization has to be performed independently at each node.

In our design, we will look for quantization bins in a given node that can be *merged* without affecting localization. As will be discussed next, this is because the ambiguity created by the merger can be resolved once information obtained from the other nodes is taken into account. Note that this is the basic principle behind distributed source coding techniques: binning at the encoder, which can be disambiguated once side information is made available at the decoder [11, 12, 15] (in this case, quantized values from other nodes).

Merging of bins results in bit rate savings because fewer quantization indices have to be transmitted. To quantify the bit rate savings, we need to take into consideration that quantization indices will be entropy coded (in this paper, Huffman coding is used). Thus, when evaluating the possible merger of two bins, we will compute the probability of the merged bin as the sum of the probabilities of the bins merged. Suppose that Q_i^j and Q_i^k are merged into $Q_i^{\min(j,k)}$. Then, we can construct the set $S_i^{\min(j,k)}$ and compute the probability for the merged bin as follows:

$$S_i^{\min(j,k)} = S_i^j \cup S_i^k, \quad (5)$$

$$P_i^{\min(j,k)} = P_i^j + P_i^k,$$

where $P_i^j = \int_{\mathbf{x} \in A_i^j} p(\mathbf{x}) d\mathbf{x}$, $p(\mathbf{x})$ is the pdf of the source position and A_i^j is given by

$$A_i^j = \{\mathbf{x} \mid (Q_1 = \alpha_1(z_1(\mathbf{x})), \dots, Q_M = \alpha_M(z_M(\mathbf{x}))) \in S_i^j\}. \quad (6)$$

Since the encoder at node i merges Q_i^j and Q_i^k into Q_i^l with $l = \min(j, k)$, it sends the corresponding index, l to the fusion node whenever the sensor reading belongs to Q_i^j or Q_i^k . The decoder will try to determine which of the two merged bins (Q_i^j or Q_i^k in this case) actually occurred at node i . To do so, the decoder will use the information provided by the other nodes, that is, the quantization indices Q_m ($m \neq i$). Consider one particular source position $\mathbf{x} \in S$ for which node i produces Q_i^j and the remaining nodes produce a combination of $M - 1$ quantization indices $\overline{\mathbf{Q}} \in \overline{S}_i^j$. (To avoid confusion, we denote \mathbf{Q} a vector of M quantization indices and $\overline{\mathbf{Q}}$ a vector of $M - 1$ quantization indices, resp.) Then, for this \mathbf{x} there would be no ambiguity at the decoder, even if bins Q_i^j and Q_i^k were to be merged, as long as $\overline{\mathbf{Q}} \notin \overline{S}_i^k$. This follows because if $\overline{\mathbf{Q}} \notin \overline{S}_i^k$ the decoder would be able to determine that only Q_i^j is consistent with receiving $\overline{\mathbf{Q}}$. With the notation adopted earlier this leads to the following definition:

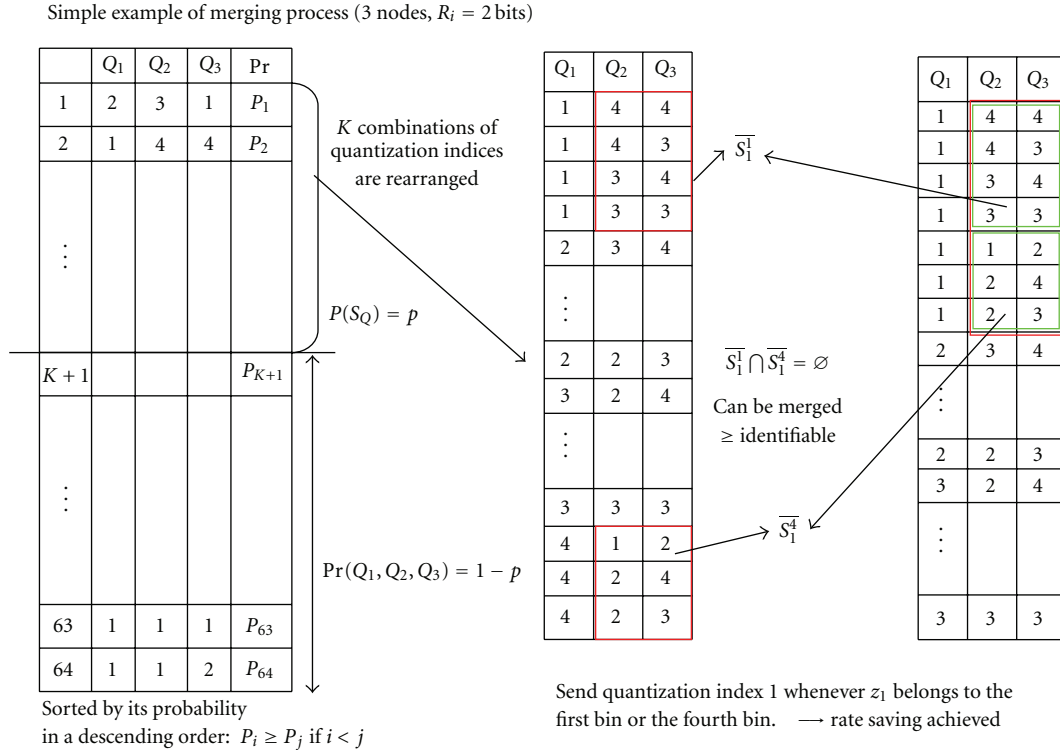


FIGURE 3: Simple example of merging process, where there are 3 nodes and each node uses a 2 bit quantizer ($Q_i \in \{1, 2, 3, 4\}$). In this case, it is assumed that $\Pr(S_M - S_Q) = 1 - p \approx 0$.

Definition 1. Q_i^j and Q_i^k are identifiable, and therefore can be merged, if and only if $\bar{S}_i^j \cap \bar{S}_i^k = \emptyset$.

Figure 3 illustrates how to merge quantization bins for a simple case, where there are 3 nodes deployed in a sensor field. It is noted that the first bin Q_1^1 (equivalently, $Q_1 = 1$) and the fourth bin Q_1^4 at node 1 can be merged since the sets \bar{S}_1^1 and \bar{S}_1^4 have no elements in common. This merging process will be repeated in the other nodes until there are no quantization bins that can be merged.

4. Quantization Schemes

As mentioned in the previous section, there will be redundancy in M -tuples after quantization which can be eliminated by our merging technique. However, we can also attempt to reduce the redundancy during quantizer design before the encoding of the bins is performed. Thus, it would be worth considering the effect of selection of a given quantization scheme on system performance when the merging technique is employed. In this section, we consider three schemes as follows.

(i) *Uniform quantizers.* Since they do not utilize any statistics about the sensor readings for quantizer design, there will be no reduction in redundancy by the quantization scheme. Thus only the merging technique plays a role in improving the system performance.

(ii) *Lloyd quantizers.* Using the statistics about the sensor reading z_i available at node i , the i th quantizer α_i is designed using the generalized Lloyd algorithm [16] with the cost function $|z_i - \hat{z}_i|^2$ which is minimized in an iterative fashion. Since each node consider only the information available to it during quantizer design, there will still exist much redundancy after quantization which the merging technique can attempt to reduce.

(iii) *Localization specific quantizers (LSQs) proposed in [7].* While designing a quantizer at node i , we can take into account the effect of quantized sensor readings at other nodes on the quantizer design by introducing the localization error in a new cost function, which will be minimized in an iterative manner. (The new cost function to be minimized is expressed as the Lagrangian functional $|z_i - \hat{z}_i|^2 + \lambda \|x - \hat{x}\|^2$. The topic of quantizer design in distributed setting goes beyond the scope of this work. See [7, 8] for detailed information.) Since the correlation between sensor readings is exploited during quantizer design, LSQ along with our merging technique will show the best performance of all.

We will discuss the effect of quantization and encoding on the system performance based on experiments for an acoustic amplitude sensor system in Section 9.1.

5. Proposed Encoding Algorithm

In general, there will be multiple pairs of identifiable quantization bins that can be merged. Often, all candidate

identifiable pairs cannot be merged simultaneously; that is, after a pair has been merged, other candidate pairs may become nonidentifiable. In what follows, we propose algorithms to determine in a sequential manner which pairs should be merged.

In order to minimize the total rate consumed by M nodes, an optimal merging technique should attempt to reduce the overall entropy as much as possible, which can be achieved by (1) merging high probability bins together and (2) merging as many bins as possible. It should be observed that these two strategies cannot be pursued simultaneously. This is because high probability bins (under our assumption of uniform distribution of the source position) are large and thus merging large bins tends to result in fewer remaining merging choices (i.e., a larger number of identifiable bin pairs may become nonidentifiable after two large identifiable bins have been merged). Conversely, a strategy that tries to maximize the number of merged bins will tend to merge many small bins, leading to less significant reductions in overall entropy. In order to strike a balance between these two strategies, we define a metric, W_i^j , attached to each quantization bin

$$W_i^j = P_i^j - \gamma |S_i^j|, \quad (7)$$

where $\gamma \geq 0$. This is a weighted sum of the bin probability and the number of the combinations of M -tuples that include Q_i^j . If P_i^j is large the corresponding bin would be a good candidate for merging under criterion (1) whereas a small value of $|S_i^j|$ will indicate a good choice under criterion (2). In our proposed procedure, for a suitable value of γ , we will seek to prioritize the merging of those identifiable bins having the largest total weighted metric. This will be repeated iteratively until there are no identifiable bins left. The selection of γ can be heuristically made so as to minimize the total rate. For example, several different γ 's could be evaluated in (7) to first determine its applicable range which will be then searched to find a proper value of γ . Clearly, γ depends on the application.

The proposed *global merging algorithm* is summarized as follows.

Step 1. Set $F(i, j) = 0$, where $i = 1, \dots, M$; $j = 1, \dots, L_i$, indicating that none of the bins, Q_i^j , have been merged yet.

Step 2. Find $(a, b) = \arg \max_{(i,j) | F(i,j)=0} (W_i^j)$, that is, we search over all the nonmerged bins for the one with the largest metric W_a^b .

Step 3. Find $Q_a^c, c \neq b$ such that $W_a^c = \max_{j \neq b} (W_a^j)$, where the search for the maximum is done only over the bins identifiable with Q_a^b at node a and go to Step 4. If there are no bins identifiable with Q_a^b , set $F(a, b) = 1$, indicating the bin Q_a^b is no longer involved in the merging process. If $F(i, j) = 1$, for all i, j , stop; otherwise, go to Step 2.

Step 4. Merge Q_a^b and Q_a^c to $Q_a^{\min(b,c)}$ with $S_a^{\min(b,c)} = S_a^b \cup S_a^c$. Set $F(a, \max(b, c)) = 1$. Go to Step 2.

In the proposed algorithm, the search for the maximum of the metric is done for the bins of all nodes involved. However, different approaches can be considered for the search. These are explained as follows.

Method 1 (Complete sequential merging). In this method, we process one node at a time in a specified order. For each node, we merge the maximum number of bins possible before proceeding to the next node. Merging decisions are not modified once made. Since we exhaust all possible mergers in each node, after scanning all the nodes no more additional mergers are possible.

Method 2 (Partial sequential merging). In this method, we again process one node at a time in a specified order. For each node, among all possible bin mergers, the best one according to a criterion is chosen (the criterion could be entropy based and e.g., (7) is used in this paper) and after the chosen bin is merged we proceed to the next node. This process is continued until no additional mergers are possible in any node. This may require multiple passes through the set of nodes.

These two methods can be easily implemented with minor modifications to our proposed algorithm. Notice that the final result of the encoding algorithm will be M merging tables, each of which has the information about which bins can be merged at each node in real operation. That is, each node will merge the quantization bins using the merging table stored at the node and will send the merged bin to the fusion node which then tries to determine which bin actually occurred via the decoding process using M merging tables and S_Q .

5.1. Incremental Merging. The complexity of the above procedures is a function of the total number of quantization bins, and thus of the number of the nodes involved. These approaches could potentially be complex for large sensor fields. We now show that incremental merging is possible; that is, we can start by performing the merging based on a subset consisting of N sensor nodes, $N < M$, and it can be guaranteed that the merging decisions that were valid when N nodes were considered will remain valid even when all M nodes are taken into account. To see this, suppose that Q_i^j and Q_i^k are identifiable when only N nodes are considered. From Definition 1, $\overline{S_i^j(N)} \cap \overline{S_i^k(N)} = \emptyset$, where N indicates the number of nodes involved in the merging process. Note that since every element $\mathbf{Q}^j(M) = (Q_1, \dots, Q_N, Q_{N+1}, \dots, Q_M) \in S_i^j(M)$ (In this section, we denote by $\mathbf{Q}^j(M)$ an element $(Q_1, \dots, Q_i = j, \dots, Q_M) \in S_i^j(M)$. Later, it will be also used to denote an j th element in S_Q in Section 8 without confusion) is constructed by concatenating $M - N$ indices Q_{N+1}, \dots, Q_M with the corresponding element, $\mathbf{Q}^j(N) = (Q_1, \dots, Q_N) \in S_i^j(N)$, we have that $\mathbf{Q}^j(M) \neq \mathbf{Q}^k(M)$ if $\mathbf{Q}^j(N) \neq \mathbf{Q}^k(N)$. By the property of the intersection operator \cap , we can claim that $\overline{S_i^j(M)} \cap \overline{S_i^k(M)} = \emptyset$ for all $M \geq N$, implying that Q_i^j and Q_i^k are still identifiable even when we consider M nodes.

Thus, we can start the merging process with just two nodes and continue to do further merging by adding one node (or a few) at a time without change in previously merged bins. When many nodes are involved, this would lead to significant savings in computational complexity. In addition, if some of the nodes are located far away from the nodes being added (i.e., the dynamic ranges of their quantizers do not overlap with those of the nodes being added), they can be skipped for further merging without loss of merging performance.

6. Extension of Identifiability: p -Identifiability

Since for real operating conditions, there exist measurement noise ($w_i \neq 0$) and/or parameter mismatches, it is computationally impractical to construct the set S_Q satisfying the assumption of $\Pr[\mathbf{Q} \in S_Q] = 1$ under which the merging algorithm was derived in Section 3. Instead, we construct $S_Q(p)$ such that $\Pr[\mathbf{Q} \in S_Q(p)] = p (\approx 1)$ and propose an extended version of *identifiability* that allows us to still apply the merging technique under noisy situations. With this consideration, Definition 1 can be extended as follows.

Definition 2. Q_i^j and Q_i^k are p -identifiable, and therefore can be merged, if and only if $\overline{S_i^j(p)} \cap \overline{S_i^k(p)} = \emptyset$, where $\overline{S_i^j(p)}$ and $\overline{S_i^k(p)}$ are constructed from $S_Q(p)$ as $\overline{S_i^j}$ from S_Q in Section 2. Obviously, to maximize the rate gain achievable by the merging technique, we need to construct $S_Q(p)$ as small as possible given p . Ideally, we can build the set $S_Q(p)$ by collecting the M -tuples with high probability although it would require huge computational complexity especially when many nodes are involved at high rates. In this work, we suggest following the procedure stated below for construction of $S_Q(p)$ with reduced complexity.

Step 1. Compute the interval $I_{z_i}(x)$ such that $\mathbf{P}(z_i \in I_{z_i}(x) | \mathbf{x}) = p^{1/M} = 1 - \beta$, for all i . Since $z_i \sim N(f_i, \sigma_i^2)$, where $f_i = f(x, x_i, \mathbf{P}_i)$ in (1), we can construct the interval that is symmetric with respect to f_i ; that is, $I_{z_i}(x) = [f_i - z_{\beta/2}, f_i + z_{\beta/2}]$, so that $\prod_{i=1}^M \Pr(z_i \in I_{z_i}(x) | \mathbf{x}) = p$. Notice that $z_{\beta/2}$ is determined by σ_i and β (not a function of x). For example, if $(1 - \beta) = 0.99$, $z_{\beta/2}$ is given by $3\sigma_i$ and $p = (1 - \beta)^M = 0.95$ with $M = 5$.

Step 2. From M intervals $I_{z_i}(x), i = 1, \dots, M$, we generate possible M -tuples $\mathbf{Q} = [Q_1, \dots, Q_M]$ satisfying that $Q_i \cap I_{z_i} \neq \emptyset$, for all i . Denote by $S_Q(x)$ a set containing such M tuples. It is noted that the process of generating M -tuples from M intervals is deterministic, given M quantizers. (Simple programming allows us to generate M -tuples from M intervals. For example, suppose that $M = 3$ and $I_{z_1} = [1.2 \ 2.3]$, $I_{z_2} = [2.7 \ 3.3]$, and $I_{z_3} = [1.8 \ 3.1]$ are computed given x in Step 1. Pick an M -tuple $\mathbf{Q} \in S_M$ with $Q_1 = [1.5 \ 2.2]$, $Q_2 = [2.5 \ 3.1]$, and $Q_3 = [2.1 \ 2.8]$. Then, we determine whether or not $\mathbf{Q} \in S_Q(x)$ by checking $Q_i \cap I_{z_i} \neq \emptyset$, for all i . In this example, we have $\mathbf{Q} \in S_Q(x)$.)

Step 3. Construct $S_Q(p) = \bigcup_{\mathbf{x} \in S} S_Q(x)$. We have $\Pr(\mathbf{Q} \in S_Q(p)) = E_{\mathbf{x}}[\Pr(\mathbf{Q} \in S_Q(p) | \mathbf{x})] \approx E_{\mathbf{x}}[\prod_{i=1}^M \Pr(z_i \in I_{z_i}(x) | \mathbf{x})] = p$.

As β approaches 1, $S_Q(p)$ will be asymptotically reduced to S_Q , the set constructed in a noiseless case. It should be mentioned that this procedure provides a tool that enables us to change the size of $S_Q(p)$ by simply adjusting β . Obviously, computation of $\Pr(\mathbf{Q} | \mathbf{x})$ is unnecessary.

Notice that all the merged bins are p -identifiable (or identifiable) at the fusion node as long as the M -tuple to be encoded belongs to $S_Q(p)$ (or S_Q). In other words, decoding errors will be generated when elements in $S_M - S_Q(p)$ occur and there will be tradeoff between rate savings and decoding errors. If we choose p to be as small as possible, yielding a small set $S_Q(p)$, we can achieve good rate savings at the expense of large decoding error (equivalently, $\Pr[\mathbf{Q} \in S_M - S_Q(p)]$ large), which could lead to degradation of localization performance. Handling of decoding errors will be discussed in Section 7.

7. Decoding of Merged Bins and Handling Decoding Errors

In the decoding process, the fusion node will first decompose the received M -tuple \mathbf{Q}_r into the possible M -tuples, $\mathbf{Q}_{D_1}, \dots, \mathbf{Q}_{D_K}$ by using the M merging tables (see Figure 4). Note that the merging process is done offline in a centralized manner. In real operation, each node stores its merging table which is constructed from the proposed merging algorithm and used to perform the encoding and the fusion node uses $S_Q(p)$ and M merging tables to do the decoding. Revisit the simple case in Figure 3. According to node 1's merging table, Q_1^1 and Q_1^4 can be merged into Q_1^1 , implying that node 1 will transmit Q_1^1 to the fusion node whenever z_1 belongs to Q_1^1 or Q_1^4 . Suppose that the fusion node receives $\mathbf{Q}_r = (1, 2, 4)$. Then, it decomposes $(1, 2, 4)$ into $(1, 2, 4)$ and $(4, 2, 4)$ by using node 1's merging table. This decomposition will be performed for the other $M - 1$ merging tables. Note that $(1, 2, 4)$ is discarded since it does not belong to $S_Q(p)$, implying that Q_1^4 actually occurred at node 1.

Suppose that we have a set of K M -tuples, $S_D = \{\mathbf{Q}_{D_1}, \dots, \mathbf{Q}_{D_K}\}$ decomposed from \mathbf{Q}_r via M merging tables. Then, clearly, $\mathbf{Q}_r \in S_D$ and $\mathbf{Q}_t \in S_D$, where \mathbf{Q}_t is the true M -tuple before encoding (see Figure 4). Notice that if $\mathbf{Q}_t \in S_Q(p)$, then all merged bins would be identifiable at the fusion node; that is, after decomposition, there is only one decomposed M -tuple, \mathbf{Q}_t belonging to $S_Q(p)$. (As the decomposition is processed, all the decomposed M -tuples except \mathbf{Q}_t will be discarded since they do not belong to $S_Q(p)$.) and we declare decoding successful. Otherwise, we declare decoding errors and apply the decoding rules which will be explained in the following subsections, to handle those errors. Since the decoding error occurs only when $\mathbf{Q}_t \notin S_Q(p)$, the decoding error probability will be less than $1 - p$.

It is observed that since the decomposed M -tuples are produced via the M merging tables from \mathbf{Q}_t , it is very likely that $\Pr(\mathbf{Q}_{D_k}) \ll \Pr(\mathbf{Q}_t)$, where $\mathbf{Q}_{D_k} \neq \mathbf{Q}_t, k = 1, \dots, K$.

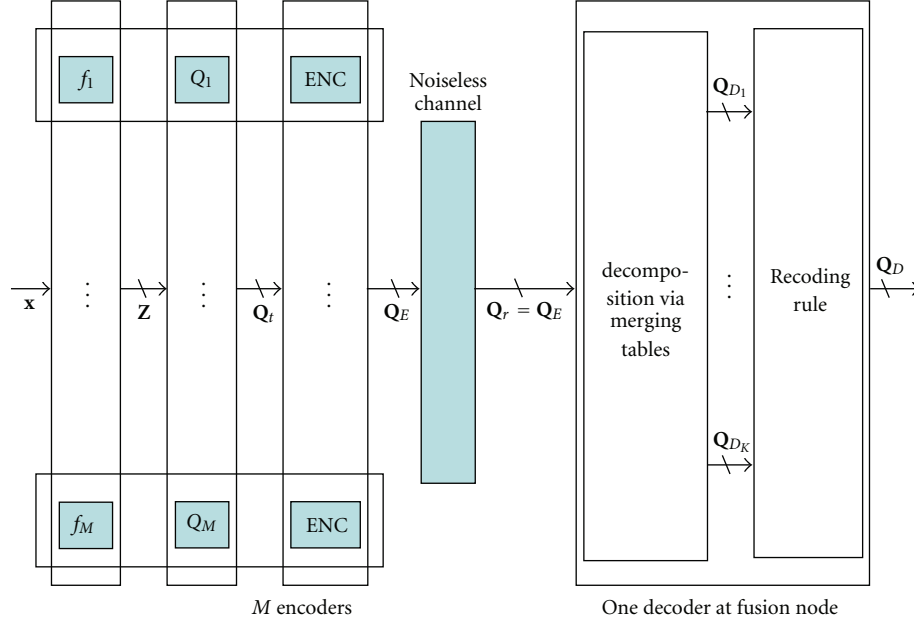


FIGURE 4: Encoder-decoder diagram: the decoding process consists of decomposition of the encoded M -tuple \mathbf{Q}_E and decoding rule of computing the decoded M -tuple \mathbf{Q}_D which will be forwarded to the localization routine.

In other words, since the encoding process merges the quantization bins whenever any M -tuples that contain either of them are very unlikely to happen at the same time, the M -tuples \mathbf{Q}_{D_k} ($\neq \mathbf{Q}_t$) tend to take very low probability.

7.1. Decoding Rule 1: Simple Maximum Rule. Since the received M -tuple \mathbf{Q}_r has ambiguity produced by encoders at each node, the decoder at fusion node should be able to find the true M -tuple by using appropriate decoding rules. As a simple rule, we can take the M -tuple (out of $\mathbf{Q}_{D_1}, \dots, \mathbf{Q}_{D_K}$) that is most likely to happen. Formally,

$$\mathbf{Q}_D = \arg \max_k \Pr[\mathbf{Q}_{D_k}], \quad k = 1, \dots, K, \quad (8)$$

where \mathbf{Q}_D is the decoded M -tuple which will be forwarded to the localization routine.

7.2. Decoding Rule 2: Weighted Decoding Rule. Instead of choosing only one decoded M -tuple, we can treat each decomposed M -tuple as a candidate for the decoded M -tuple, \mathbf{Q}_D with its corresponding weight obtained from the likelihood. That is, we can view \mathbf{Q}_{D_k} as one decoded M -tuple with weight $W_k = \Pr[\mathbf{Q}_{D_k}] / \sum_{l=1}^K \Pr[\mathbf{Q}_{D_l}]$, $k = 1, \dots, K$. It should be noted that the weighted decoding rule should be used along with the localization routine as follows:

$$\hat{x} = \sum_{k=1}^K \hat{x}_k W_k \quad k = 1, \dots, K, \quad (9)$$

where \hat{x}_k is the estimated source location assuming $\mathbf{Q}_D = \mathbf{Q}_{D_k}$. For simplicity, we can take a few dominant M -tuples

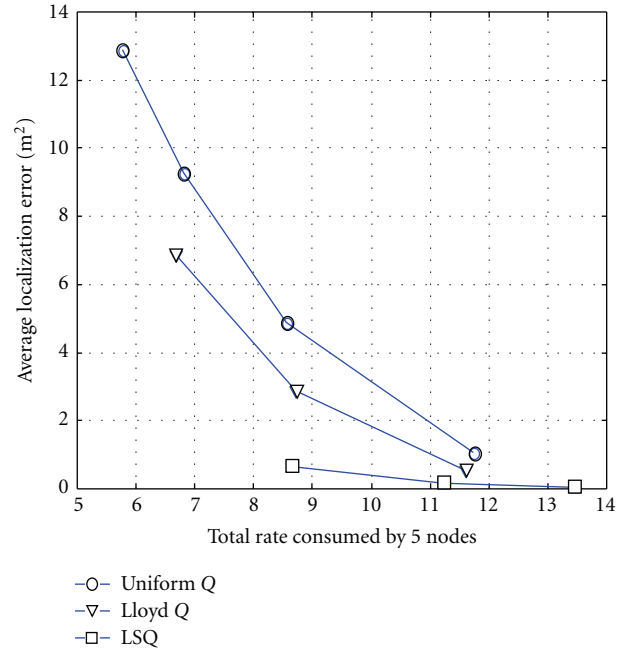


FIGURE 5: Average localization error versus total rate R_M for three different quantization schemes with distributed encoding algorithm. Average rate savings is achieved by the distributed encoding algorithm (global merging algorithm).

for the weighted decoding and localization

$$\hat{x} = \sum_{k=1}^L \hat{x}_{(k)} W_{(k)} \quad k = 1, \dots, L, \quad (10)$$

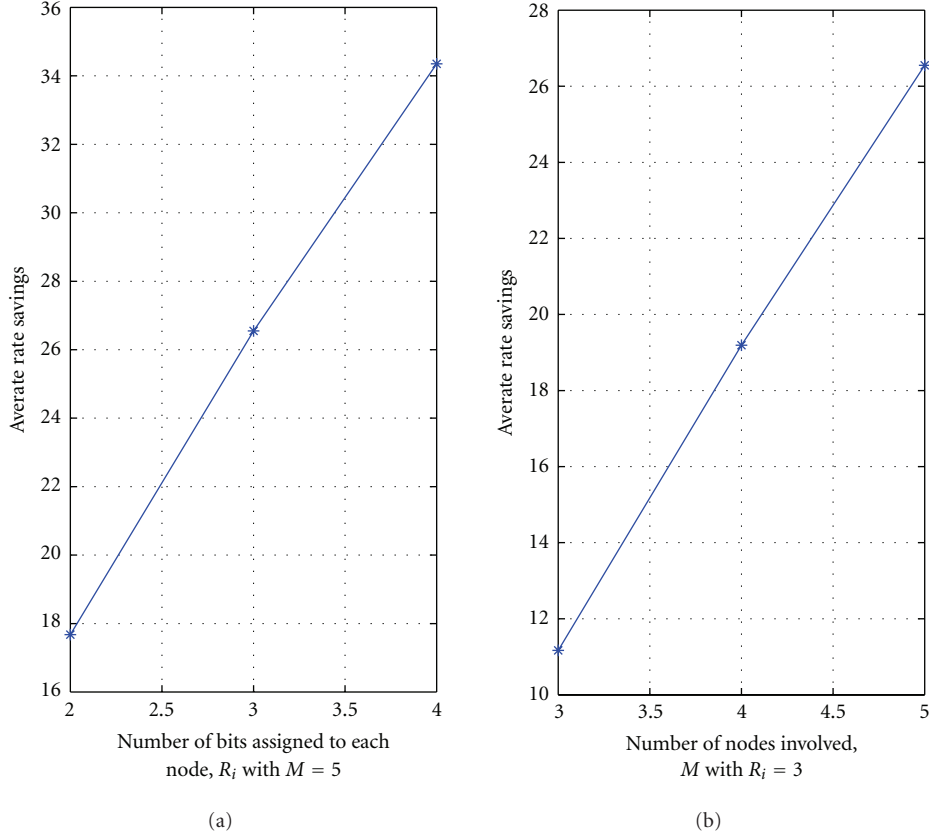


FIGURE 6: Average rate savings achieved by the distributed encoding algorithm (global merging algorithm) versus number of bits, R_i with $M = 5$ (left) and number of nodes with $R_i = 3$ (right).

where $W_{(k)}$ is the weight of $\mathbf{Q}_{D_{(k)}}$ and $\Pr[\mathbf{Q}_{D_{(i)}}] \geq \Pr[\mathbf{Q}_{D_{(j)}}]$ if $i < j$. Typically, $L (< K)$ is chosen as a small number (e.g., $L = 2$ in our experiments). Note that the weighted decoding rule with $L = 1$ is equivalent to the simple maximum rule in (8).

8. Application to Acoustic Amplitude Sensor Case

As an example of the application, we consider the acoustic amplitude sensor system, where an energy decay model of sensor signal readings proposed in [4] is used for localization. The energy decay model was verified by the field experiment in [4] and was also used in [9, 13, 17].) This model is based on the fact that the acoustic energy emitted omnidirectionally from a sound source will attenuate at a rate that is inversely proportional to the square of the distance in free space [18]. When an acoustic sensor is employed at each node, the signal energy measured at node i over a given time interval k , and denoted by z_i , can be expressed as follows:

$$z_i(x, k) = g_i \frac{a}{\|x - x_i\|^\alpha} + w_i(k), \quad (11)$$

where the parameter vector \mathbf{P}_i in (1) consists of the gain factor of the i th node g_i , an energy decay factor α , which is approximately equal to 2 in free space, and the source

signal energy a . The measurement noise term $w_i(k)$ can be approximated using a normal distribution, $N(0, \sigma_i^2)$. In (11), it is assumed that the signal energy, a , is uniformly distributed over the range $[a_{\min} \ a_{\max}]$.

In order to perform distributed encoding at each node, we first need to obtain the set S_Q , which can be constructed from (3) as follows:

$$S_Q = \left\{ (Q_1, \dots, Q_M) \mid \exists \mathbf{x} \in S, Q_i = \alpha_i \left(g_i \frac{a}{\|x - x_i\|^\alpha} + w_i \right) \right\}, \quad (12)$$

where the i th sensor reading $z_i(x)$ is expressed by the sensor model $g_i(a/\|x - x_i\|^\alpha)$, and the measurement noise, w_i .

When the signal energy a is known, and there is no measurement noise ($w_i = 0$), it would be straightforward to construct the set S_Q . That is, each element in S_Q corresponds to one region in sensor field which is obtained by computing the intersection of M ring-shaped areas (see Figure 2). For example, using an j th element $\mathbf{Q}^j = (Q_1, \dots, Q_M)$ in S_Q , we can compute the corresponding intersection A^j as follows:

$$A_i = \left\{ x \mid g_i \frac{a}{\|x - x_i\|^\alpha} \in Q_i, x \in S \right\}, \quad i = 1, \dots, M, \quad (13)$$

$$A^j = \bigcap_i^M A_i.$$

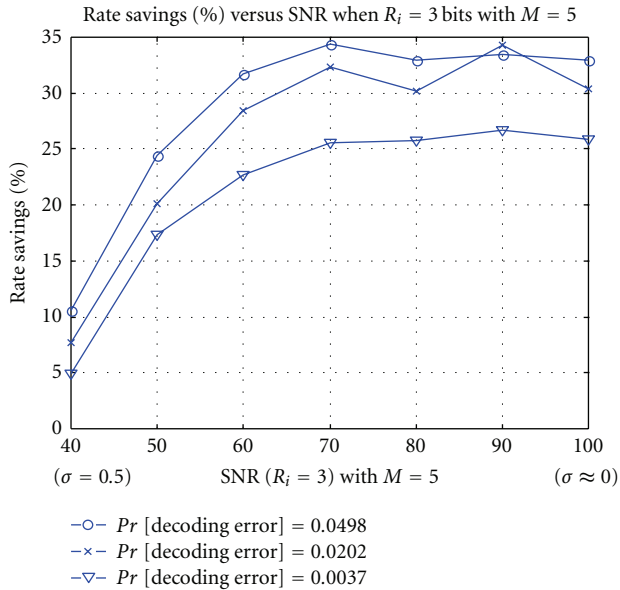


FIGURE 7: Rate savings achieved by the distributed encoding algorithm (global merging algorithm) versus SNR (dB) with $R_i = 3$ and $M = 5$. $\sigma^2 = 0, \dots, 0.5^2$.

TABLE 1: Total rate, R_M in bits (rate savings) achieved by various merging techniques.

R_i	Method 1	Method 2	Method 3
2	9.4 (8.7%)	9.4 (8.7%)	9.10 (11.6%)
3	11.9 (20.6%)	12.1 (19.3%)	11.3 (24.6%)
4	13.7 (31.1%)	14.1 (29.1%)	13.6 (31.6%)

Since the nodes involved in localization of any given source generate the same M -tuple, the set S_Q will be computed deterministically and we have $\Pr[\mathbf{Q} \in S_Q] = 1$. Thus, using S_Q , we can apply our merging technique to this case and achieve significant rate savings without any degradation of localization accuracy (no decoding error).

However, measurement noise and/or unknown signal energy will make this problem complicated by allowing random realizations of M -tuples generated by M nodes for any given source location. For this case, we construct $S_Q(p)$ by following the procedure in Section 6 and apply our decoding rules explained in Section 7 to handle decoding errors.

9. Experimental Results

The distributed encoding algorithm described in Section 5 is applied to the system, where each node employs an acoustic amplitude sensor model given by (11) for source localization. The experimental results are provided in terms of average localization error. (Clearly, the localization error would be affected by the estimators employed at the fusion node. The estimation algorithms go beyond the scope of this work. For detailed information, see [9].) $E\|\mathbf{x} - \hat{\mathbf{x}}\|^2$ and rate savings (%) computed by $((R_T - R_M)/R_T) \times 100$, where R_T is the rate

TABLE 2: Total rate R_M in bits (rate savings) achieved by distributed encoding algorithm (global merging technique). The rate savings is averaged over 20 different node configurations, where each node uses LSQ with $R_i = 3$.

M	Total rate R_M in bits (rate savings)
12	17.3237 (51.56%)
16	20.7632 (56.45%)
20	23.4296 (60.69%)

consumed by M nodes when only the independent entropy coding (Huffman coding) is used after quantization and R_M is the rate by M nodes when the merging technique is applied to quantized data before the entropy coding. We assume that each node uses LSQ described in Section 4 (for further details, refer to [7]) except for the experiments where otherwise stated.

9.1. Distributed Encoding Algorithm: Noiseless Case. It is assumed that each node can measure the known signal energy without measurement noise. Figure 5 shows the overall performance of the system for each quantization scheme. In this experiment, 100 different 5-node configurations were generated in a sensor field $10 \times 10 \text{ m}^2$. For each configuration, a test set of 2000 random source locations was used to obtain sensor readings, which are then quantized by three different quantizers, namely, uniform quantizers, Lloyd quantizers, and LSQs. The average localization error and total rate R_M are averaged over 100 node configurations. As expected, the overall performance for LSQ is the best of all since the total reduction in redundancy can be maximized when the application-specific quantization such as LSQ and the distributed encoding are used together.

Our encoding algorithm with the different merging techniques outlined in Section 5 is applied for comparison, and the results are provided in Table 1. Methods 1 and 2 are as described in Section 5, and Method 3 is the global merging algorithm discussed in that section. We can observe that even with relative low rates (4 bits per node) and a small number of nodes (only 5) significant rate gains (over 30%) can be achieved with our merging technique.

The encoding algorithm was also applied to many different node configurations to characterize the performance. In this experiment, 500 different node configurations were generated for each $M (= 3, 4, 5)$ in a sensor field $10 \times 10 \text{ m}^2$. The global merging technique has been applied to obtain the rate savings. In computing the metric in (7), the source distribution is assumed to be uniform. The average rate savings is plotted by varying M and R_i in Figure 6. Clearly, the better rate savings is achieved with larger M and/or at higher rate since there exists more redundancy expressed as $|S_M - S_Q|$, as more nodes become involved at higher rate.

Since there are a large number of nodes in typical sensor networks, our distributed algorithms have been applied to the system in a larger sensor field ($20 \times 20 \text{ m}^2$). In this experiment, 20 different node configurations are generated for each $M (= 12, 16, 20)$. Note that the node density for $M = 20$ in $20 \times 20 \text{ m}^2$ is equal to $20/(20 \times 20) = 0.05$ which

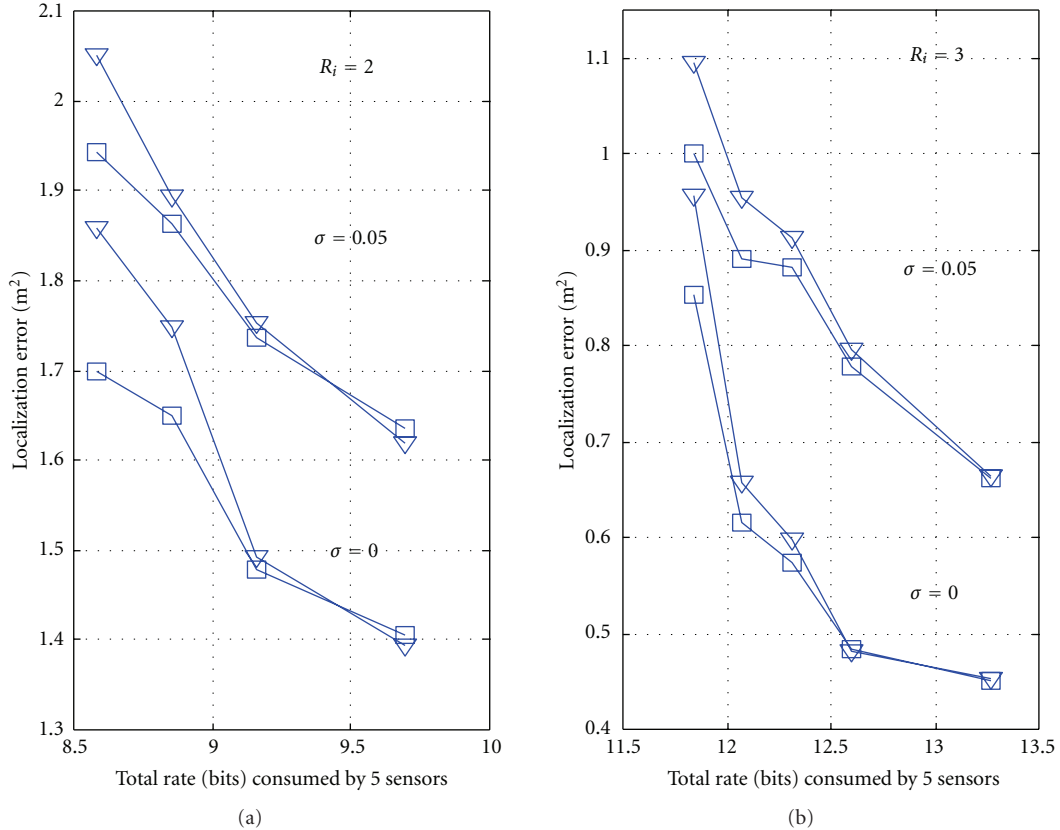


FIGURE 8: Average localization error versus total rate R_M achieved by the distributed encoding algorithm (global merging algorithm) with simple maximum decoding and weighted decoding, respectively. Total rate increases by changing p from 0.8 to 0.95 and weighted decoding is conducted with $L = 2$. Solid line + \square : weighted decoding. Solid line + ∇ : simple maximum decoding.

is also the node density for the case of $M = 5$ in $10 \times 10 \text{ m}^2$. In Table 2, it is worth noting that the system with a larger number of nodes outperforms the system with a smaller number of nodes ($M = 3, 4, 5$) although the node density is kept the same. This is because the incremental property of the merging technique allows us to find more identifiable bins at each node.

9.2. Encoding with p -Identifiability and Decoding Rules: Noisy Case. The distributed encoding algorithm with p -identifiability described in Section 6 was applied to the case, where each node collects noise-corrupted measurements of unknown source signal energy. First, assuming known signal energy, we checked the effect of measurement noise on the rate savings, and thus the decoding error by varying the size of $S_Q(p)$. Note that as p becomes increased, the total rate R_M tends to be increased since small rate gain is achieved with $S_Q(p)$ large. In this experiment, the variance of measurement noise, σ^2 , varies from 0 to 0.5^2 and for each σ^2 , a test set of 2000 source locations was generated with $a = 50$. Figure 7 illustrates that good rate savings can be still achieved in a noisy situation by allowing small decoding errors. It can be noted that better rate savings can be achieved at higher SNR (Note that for practical vehicle target, the SNR is often much higher than 40 dB and a typical value of the variance of

measurement noise σ^2 is 0.05^2 [4, 13].) and/or with larger decoding errors allowed ($\Pr[\text{decoding error}] < 0.05$ in this experiments).

For the case of unknown signal energy, where we assume that $a \in [a_{\min} \ a_{\max}] = [0 \ 100]$, we constructed $S_Q(p) = \bigcup_{k=1}^{L_a} S_Q(a_k)$ with $\Delta a = a_{k+1} - a_k = (a_{\max} - a_{\min})/L_a = 0.5$ by varying $p = 0.8, \dots, 0.95$, where $S_Q(a_k)$ is constructed when $a = a_k$ using the procedure in Section 6. Using $S_Q(p)$, we applied the merging technique with p -identifiability to evaluate the performance (rate savings versus localization error). In the experiment, a test set of 2000 samples is generated from uniform priors for $p(x)$ and $p(a)$ with each noise variance ($\sigma = 0$ and 0.05). In order to deal with decoding errors, two decoding rules in Section 7 were applied. In Figure 8, the performance curves for two decoding rules were plotted for comparison. As can be seen, the weighted decoding rule performs better than the simple maximum rule since the former takes into account the effect of the other decomposed M -tuples on localization accuracy by adjusting their weights. It is also noted that when decoding error is very low (equivalently, $p \approx 1$), both of them show almost the same performance.

To see how much gain we can obtain from the encoding under noisy situations, we compared this to the system which uses only the entropy coding without applying the merging

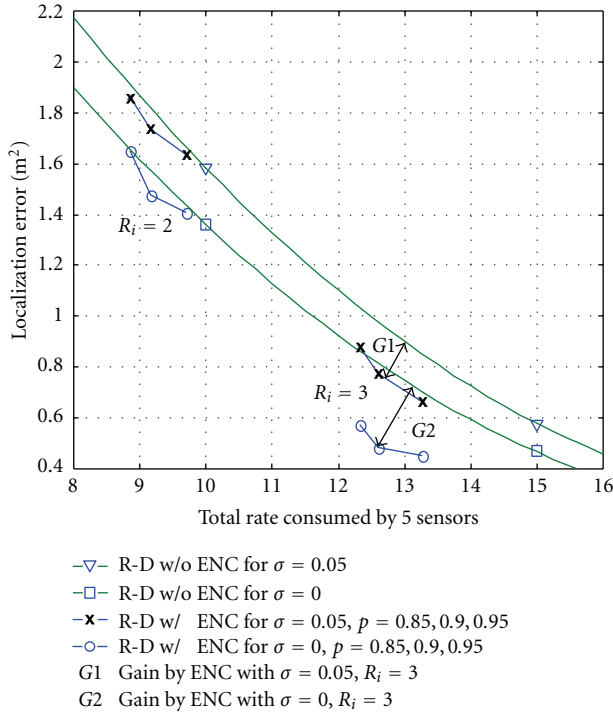


FIGURE 9: Average localization error versus total rate, R_M achieved by the distributed encoding algorithm (global merging algorithm) with $R_i = 3$ and $M = 5$. $\sigma = 0, 0.05$. $S_Q(p)$ is varied from $p = 0.85, 0.9, 0.95$. Weighted decoding with $L = 2$ is applied in this experiment.

technique. In Figure 9, the performance curves (R-D curves) are plotted with $p = 0.85, 0.9$ and 0.95 for $\sigma = 0$ and 0.05 . It should be noted that we can determine the size of $S_Q(p)$ (equivalently, p) that provides the best performance from this experiment.

9.3. Performance Comparison. For the purpose of evaluation, it would be meaningful to compare our encoding technique with LSQ algorithm since both of them are optimized for source localization and can be viewed as DSC (distributed source coding) techniques which are developed as a tool to reduce the rate required to transmit data from all nodes to the sink. In Figure 10, the R-D curve for LSQ only (without our encoding technique) is plotted for comparison. It should be observed that at high rate, the encoding technique will outperform LSQ since the better rate savings will be achieved as the total rate increases.

We address the question of how our technique compares with the best achievable performance for this source localization scenario. As a bound on achievable performance we consider a system where (i) each node quantizes its measurement independently and (ii) the quantization indices generated by all nodes for a given source location are jointly coded (in our case, we use the joint entropy of the vector of measurements as the rate estimate).

Note that this is not a realistic bound because joint coding cannot be achieved unless the nodes are able to

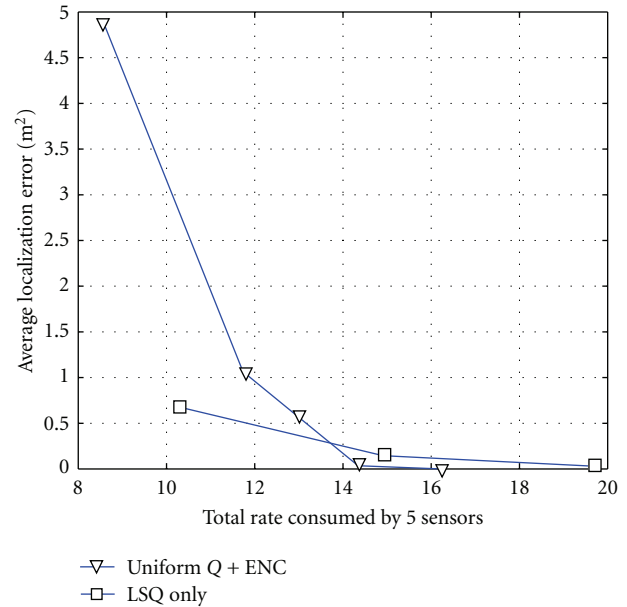


FIGURE 10: Performance comparison: Uniform quantizer equipped with distributed encoding algorithm versus LSQ only. Average localization error and total rate, R_M are averaged for 100 different 5-node configurations.

communicate before encoding. In order to approximate the behavior of the joint entropy coder via DSC techniques one would have to transmit multiple sensor readings of the source energy from each node, as the source is moving around the sensor field. Some of the nodes could send measurements that are directly encoded, while others could transmit a syndrome produced by an error correcting code based on the quantized measurements. Then, as the fusion node receives all the information from the various nodes it would be able to exploit the correlation from the measurements and approximate the joint entropy. This method would not be desirable, however, because the information in each node depends on the location of the source and thus to obtain a reliable estimate of the measurement at all nodes one would have to have measurements at a sufficient number of positions of the source. Thus, instantaneous localization of the source would not be possible. The key point here, then, is that the randomness between measurements across nodes is based on the localization of the source, which is precisely what we wish to observe.

For a 5-node configuration, the average rate per node was plotted with respect to the localization error in Figure 11, with assumption of no measurement noise ($w_i = 0$) and known signal energy. For this particular configuration we can observe a gap of less than 1 bit/node, at high rates, between the performance achieved by the distributed encoding and that achievable by the joint entropy coding when the same quantizers (LSQ) are employed. In summary, our merging technique provides substantial gain which comes close to the optimal achievable performance.

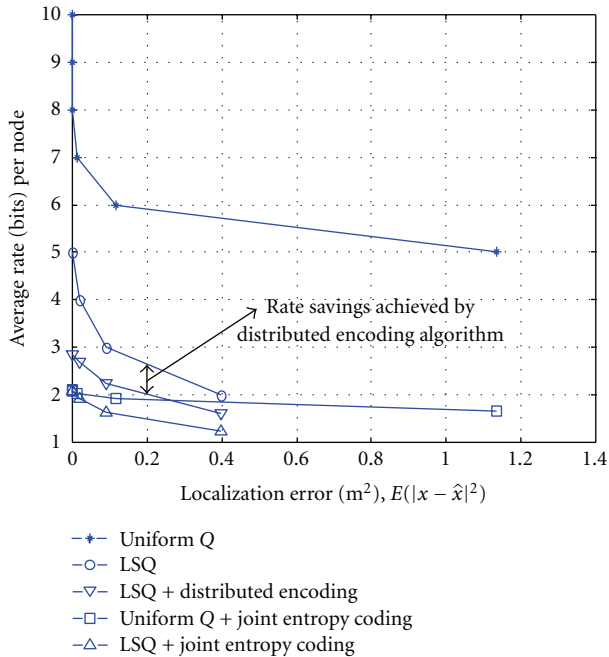


FIGURE 11: Performance comparison: distributed encoding algorithm is lower bounded by joint entropy coding.

10. Conclusion and Future Works

Using the distributed property of the quantized sensor readings, we proposed a novel encoding algorithm to achieve significant rate savings by merging quantization bins. We also developed decoding rules to deal with the decoding errors which can be caused by measurement noise and/or parameter mismatches. In the experiment, we showed that the system equipped with the distributed encoders achieved significant data compression as compared with standard systems.

So far, we have considered encoding algorithms by fixing quantizers. However, since there exists dependency between quantization and encoding of quantized data which can be exploited to obtain better performance gain, it would be worth considering a joint design of quantizers and encoders.

Acknowledgments

The authors would like to thank the anonymous reviewers for their careful reading of the paper and useful suggestions which led to significant improvements in the paper. This research has been funded in part by the Pratt & Whitney Institute for Collaborative Engineering (PWICE) at USC, and in part by NASA under the Advanced Information Systems Technology (AIST) program. The work was presented in part in IEEE International Symposium on Information Processing in Sensor Networks (IPSN), April 2005.

References

[1] F. Zhao, J. Shin, and J. Reich, “Information-driven dynamic sensor collaboration,” *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 61–72, 2002.

[2] J. C. Chen, K. Yao, and R. E. Hudson, “Source localization and beamforming,” *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 30–39, 2002.

[3] D. Li, K. D. Wong, Y. H. Hu, and A. M. Sayeed, “Detection, classification, and tracking of targets,” *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 17–29, 2002.

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

[5] J. C. Chen, K. Yao, and R. E. Hudson, “Acoustic source localization and beamforming: theory and practice,” *EURASIP Journal on Applied Signal Processing*, vol. 2003, no. 4, pp. 359–370, 2003.

[6] J. C. Chen, L. Yip, J. Elson et al., “Coherent acoustic array processing and localization on wireless sensor networks,” *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1154–1161, 2003.

[7] Y. H. Kim and A. Ortega, “Quantizer design for source localization in sensor networks,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’05)*, pp. 857–860, March 2005.

[8] Y. H. Kim, *Distributed algorithms for source localization using quantized sensor readings*, Ph.D. dissertation, USC, December 2007.

[9] Y. H. Kim and A. Ortega, “Maximum a posteriori (MAP)-based algorithm for distributed source localization using quantized acoustic sensor readings,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’06)*, pp. 1053–1056, May 2006.

[10] Y. H. Kim and A. Ortega, “Quantizer design and distributed encoding algorithm for source localization in sensor networks,” in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN ’05)*, pp. 231–238, April 2005.

[11] T. J. Flynn and R. M. Gray, “Encoding of correlated observations,” *IEEE Transactions on Information Theory*, vol. 33, no. 6, pp. 773–787, 1988.

[12] P. Ishwar, R. Puri, K. Ramchandran, and S. S. Pradhan, “On rate-constrained distributed estimation in unreliable sensor networks,” *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 765–774, 2005.

[13] J. Liu, J. Reich, and F. Zhao, “Collaborative in-network processing for target tracking,” *EURASIP Journal on Applied Signal Processing*, vol. 2003, no. 4, pp. 378–391, 2003.

[14] H. Yang and B. Sikdar, “A protocol for tracking mobile targets using sensor networks,” in *Proceedings of IEEE Workshop on Sensor Network Protocols and Applications (SNPA ’03)*, pp. 71–81, Anchorage, Alaska, USA, May 2003.

[15] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley-Interscience, New York, NY, USA, 1991.

[16] K. Sayood, *Introduction to Data Compression*, Morgan Kaufmann Publishers, San Francisco, Calif, USA, 2nd edition, 2000.

[17] A. O. Hero III and D. Blatt, “Sensor network source localization via projection onto convex sets (POCS),” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’05)*, pp. 689–692, March 2005.

[18] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice-Hall, Upper Saddle River, NJ, USA, 1996.