Wildlife Assessment using Wireless Sensor Networks

Harry Gros-Desormeaux, Philippe Hunel and Nicolas Vidot
LAMIA, Université des Antilles et de la Guyane,
Campus de Schœlcher, B.P. 7209, 97275 Schoelcher, French West Indies
France

1. Introduction

The endangered species always drew the attention of the scientific community since their disappearance would cause irreplaceable loss. To help these species to survive, their habitat is protected by the laws of environmental protection. Sometimes this protection is not enough, because their natural evolution is the main cause of their disappearance. However, to save them, it is sometimes possible to transfer them elsewhere that should be similar to their previous habitat to avoid disturbing the balance of wildlife. To model a habitat, several parameters must be of interest and are generally defined by experts. This is the case for the number of singing birds which will be studied in this paper.

Today, advances in sensor technology enable the monitoring of species and their habitat at a very low cost. Indeed, the increasing sophistication of wireless sensors bids opportunities that enable new challenges in a lot of areas, including the surveillance one. Progress in their miniaturization leads to micro-sensors of size of cubic millimeters which, used in large quantity, produce huge amounts of data. This paper promotes the use of sensors for monitoring bird endangered in their habitat. Actual methods for counting endangered birds use mainly human labor and because they are not really comprehensive leads to poor estimation. The use of sensors deployed in critical environments can help the census of these species and even generate new data on their customs.

Among the challenges that the use of the sensor technology enable, energy efficiency is the most critical for these wireless networks since battery depletion totally disables a sensor. In addition, designing algorithms for wireless networks stems from the distributed computer science domain with limited devices. Memory space and computational power are often of a magnitude less than miles than their desktop counterparts. This paper investigate the problem and proposes to approximate the number of birds by geometric means derived in a graph problem.

Our paper is organized as follows. First, Section 2 provides an overview of techniques generally used to estimate the locations of multiple sources with a unknown sensor network. Section 4 details our heuristics used to count birds. Section 5 introduces a distributed algorithm for counting birds. Experimentation confirms the effectiveness of our counting systems in Section 6. Then we conclude in Section 7 and gives an overview of our future work.

2. Previous Work

Source localization is an area of interest that has been widely studied in these recent years. A comprehensive review of incentives techniques and source localization has been written by Krim and Viberg in (Krim & Viberg, 1996) and it is not difficult to understand that problem has been of particular focus for military needs. Indeed, radar and sonars are a direct application of source localization.

Several acoustic parameters such as bandwidth, distance sensors, reverberation and thus change the way the location of the sources are handled. In addition, the algorithms of source localization depends strongly on physics and rely on the sound characteristics of waveform to calculate location sources. Waveform audio is known to be *broadband* (30Hz-15kHz) and sensors usually record the sound from *near-field sources*. The following presents some algorithms of interest which satisfy these two properties. Near-fields algorithms like close-formed ones (Smith. & Abel, 1987) use time delays between sensors location to estimate the source position. However, though they are computationally less expensive than maximum-likelihood parametric algorithms (Chen et al., 2001a), they cannot handle efficiently multiple sources (Chen et al., 2001b). Maximum-likelihood (ML) algorithms are inspired by the fact that source location information is contained in the linear phase shift of the sensor data spectrum obtained through a discrete Fourier Transform applied to the wideband data. However, ML techniques are dominated by low-cost suboptimal techniques like the well-known MUSIC algorithm (Schmidt, 1986) which leverages spectral calculus on signal and noise subspaces to find sources locations.

Unlike these approaches, we do not use the acoustic properties of the song of the bird to find its location. Indeed, we assume that our sensors are simple and only detect songs relevant to the monitored specie. Further, our sensors are wireless and rely on battery power to function. It is important to notice that our algorithms do not try to pinpoint birds, but rather estimate the number of songbirds that inhabit a region. In our case, only approximate geometric information is sufficient to establish this estimate.

3. Recognizing the birdsong

The recognition process of *birdsong* is the first part of our counting systems. Today, it is true that the performance levels made in the treatment of audio signals are high, but this requires large memory and processing power of large size which could exclude limited capacity of devices such as wireless sensors.

Recognition of species based on acoustic analysis has been widely studied in recent years and usually falls within the scope of the classification field. This is particularly the case for recognition of *bird songs*. Indeed, for a particular song, it is necessary to determine if it belongs to a specie. For example, the work of Seppo Fagerlund (Fagerlund, 2007) uses support vector machines to classify the different species of birds based on their songs. Similarly, Jim Cai et al. (Cai et al., 2007) propose a method recognition based on neural networks to find the membership of a song to a bird class. Our recognition process, inspired by the work of Rabiner (Rabiner & Wilpon, 1979), leverages the same mechanics by means of a clustering algorithm to classify the song.

Figure 1 gives an overview of our wireless counting system.

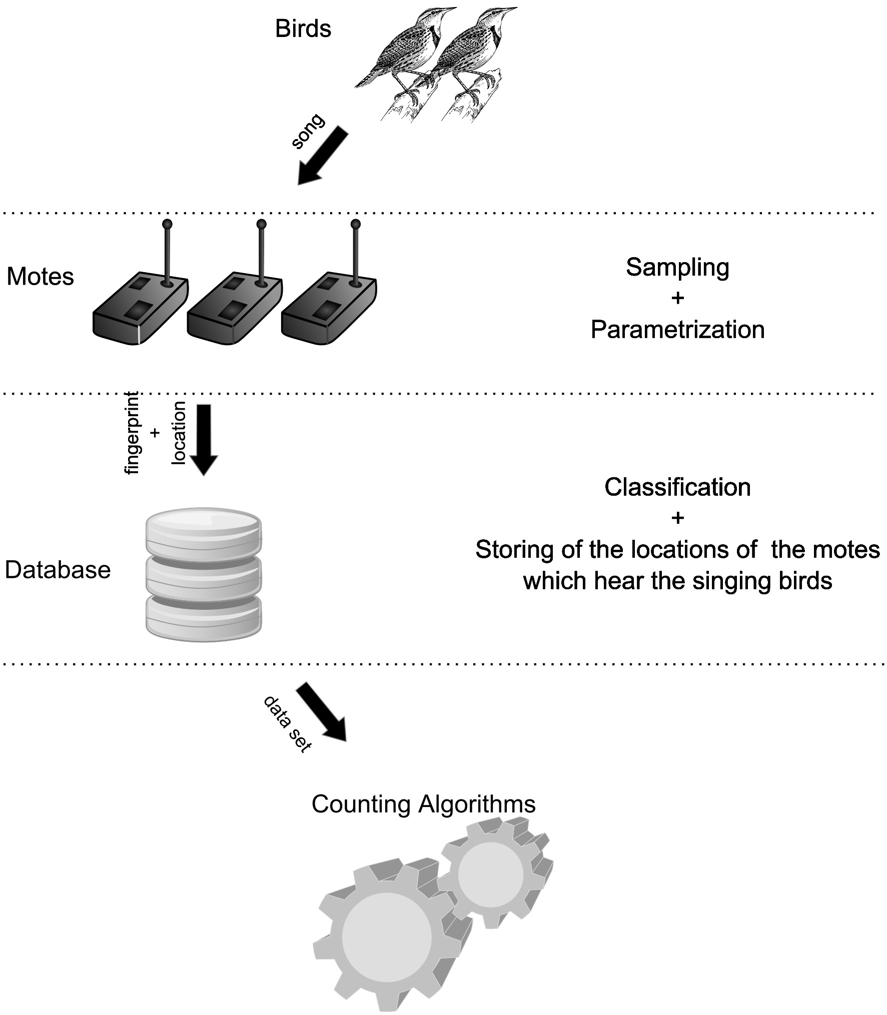

Fig. 1. The Counting System

Bird Species Recognition Using Clustering

Our classification method is twofold : a parameterization transformation process of the song in a certain fingerprint, and clustering process to determine its membership.

The parametrization process uses the songs of the birds to create a series of coefficients that describe the signal. Although various parameterization methods LPC, LPCC, PLP, dots exist, we use the MFCC Mel Frequency Cepstral Coefficient because our analysis is limited to a very limited vocabulary on limited devices. Indeed, Christopher Levy compared in (Lévy et al., 2006) different parameterization methods on small systems such as mobile phones for reduced vocabulary and have showed that the parameterization based on MFCC is much more effective for such systems.

Once the fingerprint is obtained from the parameterization process, it is added in a set with other fingerprints, themselves derived from a database containing a large number of songs of individuals known as the specie. Subsequently, a clustering algorithm (K-Means or EM) is used on all the fingerprints to determine their similarity and to create one or more clusters in which will be the *bird cluster*. For a given footprint, the problem is then to determine its membership to the bird cluster. If that's the case, data location + Mote timestamp is stored in the database for further processing counting algorithms. Our recognition results are compelling because almost all birds are classified correctly in our case.

4. The Counting Algorithm

This section is devoted to our counting heuristics inspired by the triangulation detection used by R. E. Bell to count owls in the forest (Bell, 1964). Our method differs essentially from the fact that we do not use semi-directional devices but omni-directional wireless sensors to loosely locate a birdsong. In our theoretical framework, all motes share the same characteristics building, which means they have the same (processing power, memory, battery, radius of detection, etc). Optimizing routes in wireless sensors networks here are out of concern. We only focus on the manner to detect birds in their habitat viewed as a 2D area. Further, we do not have any assumptions on the number of birds, on their movements or even their customs.
More formally, let denote $M = \{m_1, \ldots, m_n\}$ the set of all the motes which covers the habitat. Each mote has the same detection radius r. All motes can report information to the base station B which holds our counting algorithm, assuming that B is always reachable by every mote. Let $F_t : M \to \{0, 1\}$, the detection function which returns 1 if a mote m_i detects a bird, 0 otherwise at time t. The base station stores the detection array $D_t = [F_t(m_1), \ldots, F_t(m_n)]$ which reveals the detection state of each mote at time t. Note that the base stores detection arrays at a sampling rate determined empirically, that is, detection arrays D_i are stored in a data set D at the base B. Fig. 2 shows an example of motes placed on a 2D area.

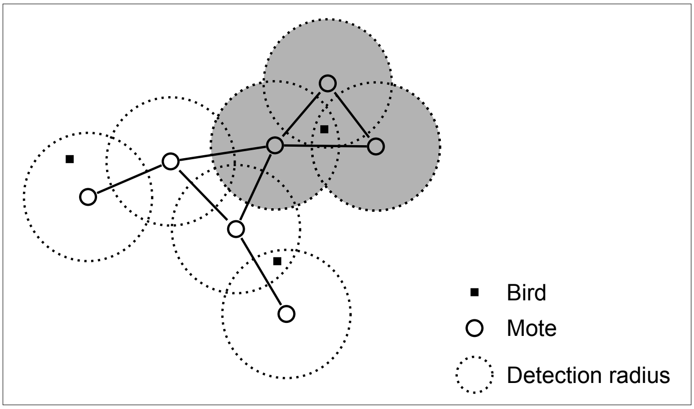

Fig. 2. Motes, birds and the hard underlying unity Chart

We propose to count one bird for all the motes which trigger at time t and for which radius of detection intersect mutually. We call such a set a Maximal Detection Set denoted MDS(N) with $N \subset M$ where N is the set of the motes which trigger at time t. The grayed area in figure 2 is a MDS. Let's denote such a subset $W = \{m \in M \mid \forall m_i, m_j \in M, r(m_i) \wedge r(m_j)\}$.

Finding the Maximum Detection Set is similar to find a maximum clique (Bomze et al., 1999). Let's see why.

A unit disk graph $G(V, E)$ is an intersection graph of disks of unit radius, that is, $\forall ij \in E$, the unit circle of center i intersects the unit circle of center j. The set of each center of these circles is called the model of the unit disk graph. This class of graph is well studied and is extensively used in the field of ad hoc networks (Kuhn et al., 2008). Indeed, UDGs (Unit Disk Graphs) can represent an ideal view of an ad hoc networks and provides strong theoretical result due to the geometric properties of the model. For example, Clark and al. (Clark et al., 1990) show that finding a maximal clique for an UDG is polynomial given its model. More recently, Raghavan and Spinrad (Raghavan & Spinrad, 2003) have shown that it is even possible to compute the maximum clique without the model in polynomial time.

Without loss of generality, let $G(V, E)$ a graph where V is the set of the motes and E, the set of edges where the edge ij exists if and only if the detection radius of mote i intersects the detection radius of mote j. Clearly, G is a unit disk graph. Unfortunately, a clique in G only gives motes which are pairwise adjacent and we are interested in motes which are mutually adjacent, that is motes which intersect mutually. We propose to alter all triangles (clique of size 3) which do not have a mutual intersection in the graph i.e we remove one edge in the triangle. As a consequence, all cliques of more than three vertices will have a mutual intersection.

Theorem 4.1. *If a graph $G(V, E)$ only has triangles formed from motes whose detection radius intersect mutually, then all motes forming a clique in G have detection radii intersecting mutually.*

Proof. By definition, all clique of size three have detection radii which intersect mutually. Now, assume that all motes clique of size n intersect mutually. Let choose such a clique that we call $S = \{m_1, \ldots, m_n\}$ and let's add a new mote m_{n+1} to S. Assume that $S + \{m_{n+1}\}$ form a clique for which some motes do not intersect mutually. Clearly, m_{n+1} form at least two proper intersections with S, and the detection radius of the mote m_{n+1} cannot intersect mutually at least with two other radii detection. But, by definition, all triangles intersect mutually which is a contradiction. □

Reichling (Reichling, 1988) uses convex programming to find the common intersection of a set of disks in $O(k)$ steps where k is the number of constraints of the convex program. Moreover, all the triangles in a graph can be computed in $O(mn)$ steps where m is the number of edges and n, the number of vertices. Thus, we can alter all triangles which do not have a common intersection in $O(kmn)$ steps. Several strategies could be used to alter a triangle. However, removing the longest edge in a triangle seems to be the most relevant one since the number of altered triangles would be reduced. Intuitively, a longest edge in a "bad" triangle is more likely to be common to another "bad" triangle. Unfortunately, the underlying unit disk graph can loose its nature since it might become a quasi-unit disk graph[1] for which the maximum clique problem is known to be NP-complete (Ceroi, 2002).

Algorithm 1 recursively constructs the maximum set of all motes which triggers at time t and removes a MDS built from this set. For each MDS removed, the number of birds iterates. This procedure is run for each detection array and the maximum number found over these detection arrays is an estimation of the number of singing birds. This algorithm complexity is bounded by the MDS search which consists in finding a clique in the unit disk graph underlying our network. Breu (Breu, 1996) has given an algorithm which find a maximum clique in a unit disk graph with complexity $O(n^{3.5} \log n)$. However, the alteration of the underlying unit disk graph leads to a NP-complete algorithm.

[1] Model which takes into account non-circular detection area

```
begin
    $L \leftarrow \emptyset$;
    foreach $d \in D$ do
        NumberOfBirds $\leftarrow 0$;
        Construct the underlying altered unit disk graph $G(V, E)$ from $d$;
        while $V \neq \emptyset$ do
            Search for a maximum clique in $G$;
            Remove this clique from $G$;
            NumberOfBirds $\leftarrow$ NumberOfBirds $+ 1$;
        Add NumberOfBirds to $L$;
    return $\max_{l \in L} l$;
end
```

Algorithm 1: The Counting Heuristic

Refining the Counting Heuristic

In the following, we suggest a little enhancement of our scheme. Indeed, we partition successive detection arrays pairwise in order to refine our estimation of the number of birds. Intuitively, the habitat is divided in such a manner that birds in a part could not have moved to another one between two instants (for each couple of detection arrays). A threshold is empirically fixed for the flight speed of the birds such that no birds can fly over that value. This leads to the decomposition of the environment in several sub-environments. Then, each sub-environment is processed with algorithm 1. For example, assume that we have 10 birds in an area. Halve this area and put 5 birds in one part, and 5 in the counterpart. Now, assume that the 5 birds in the first part sing together at time t, the other ones sing together at time $t + 1$ and these parts are too distant such that birds in one part can go in the other part between the two time steps. In that case, algorithm 1 outputs 5 birds as estimate. Our next algorithm halves the environment in two parts such that birds in two. As a consequence, we can apply algorithm 1 on each part independently and take the sum of the estimates found on each part, which gives 10 birds.

```
Data: A list of detection arrays $D = D_1, \ldots, D_m$
Result: An estimation of the number of birds in the habitat
begin
    $L \leftarrow \emptyset$;
    while $|D| > 0$ do
        Partition detection arrays $D_i$ and $D_{i+1}$ respectively in $X = \{X_1, \ldots, X_k\}$ and
        $Y = \{Y_1, \ldots, Y_k\}$;
        $Z = \emptyset$;
        for $i \leftarrow 1$ to $k$ do
            Process $X_i$ and $Y_i$ with algorithm 1 and put the maximum of the number of birds
            counted in $Z$;
        Add $\sum_{z \in Z} z$ to $L$;
    return $\max_{l \in L} l$;
end
```

Algorithm 2: The Enhanced Counting Algorithm

For sake of clarity, in algorithm 2, the number of detection arrays is even and only two successive detection arrays are partitioned. The next section presents another way to count the singing birds in their habitat. This next version is designed to be partially distributed on the motes.

5. The Swarm Counting Protocol

Our next counting method can be seen as two levels, a local and a global one. At the local level, motes cooperates sending information to count the number of singing birds in their neighborhood. At the global level, motes aggregates data to find a more accurate estimation of the number of singing birds in the habitat.

Like the technique previously described, we assume that the motes layout forms a unit disk graph. First, motes have to estimate locally how many birds had sang. Then, they send this data to the base station which derives from all the information the estimate for the number of singing birds. In our scheme, motes all have a set of rules which are the following. They are all in a passive state until some songs trigger them. When triggered, they switch to an active state and tell to their neighbors[2] that they detect a bird. Then they listen for their neighborhood during a specified time. Finally, they deduce the number of singing birds in the vicinity from their active answering neighbors, and send this number to the base station.

Local Counting

Our local counting is somewhat similar to the one in section 4. It leverages the trilateration technique to estimate a number of birds in the vicinity. All motes know their neighbors' topology and are in an initial passive state when they are waiting for signals (bird songs). Whenever a mote is triggered , it sends a signal to its neighbors and listen for whose which were triggered too. If two or more neighbors have an intersecting detection area, we assume that only one bird is counted for these motes. In figure 3, the black mote hears a bird song, asks its neighbors if they heard too and waits for their reply. Remark that the number of birds counted is the number of neighbors which are independent mutually in each neighborhood, i.e the cardinal of the maximum independent set[3] in the graph induced by the neighbors.

Global Counting

Now, assume that all motes have counted the birds in their vicinity and have sent their local count to the base station. Now, all these information have to be aggregated accordingly to find an estimate of the number of singing birds at this instant. Because, the neighborhood was used to derive the local counting, obviously, motes which are neighbors will influence each other in the counting process. So, summing up their local count can lead to an over-estimate of the number of singing birds. Note that is also the case for motes which are at distance 2, that is neighbors of neighbors in the unit disk graph, since they can share common neighbors. Therefore, only motes which are more than distant 2 each other will sum up their count. Our estimation will be the maximum number of birds which could be counted over aggregated nodes in the underlying graph.

In figure 4, the black nodes are at distant 3. So a global counting of singing birds could be four. Remark that such a counting has to be done for all set of nodes which are at more than distance 2 each other. If such a technique seems to lead to a combinatorial explosion of the

[2] As previously, neighbors are adjacent nodes in the unit disk graph

[3] The largest set of vertices which are not pairwise adjacent

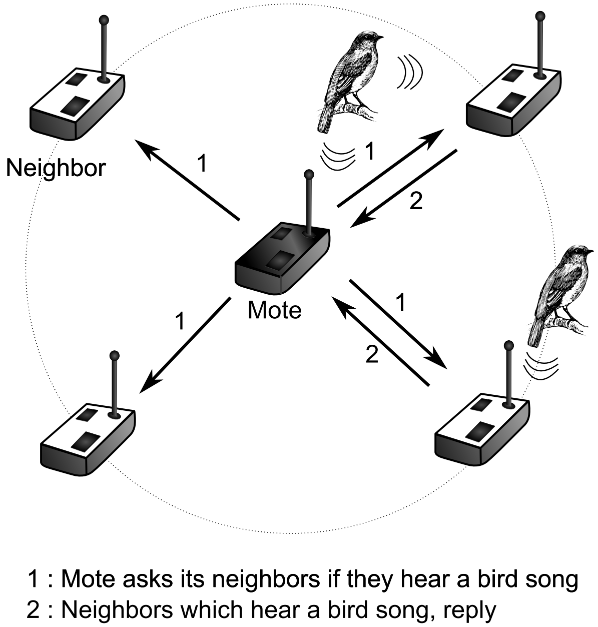
Neighbor
1
1
2
Mote
1
1
2
1 : Mote asks its neighbors if they hear a bird song
2 : Neighbors which hear a bird song, reply

Fig. 3. Motes collaboration at the local level

set of motes which can be aggregated, the underlying graph has some nice properties which allows to find the estimate in linear time.

More formally, let $N(i)$ define the neighbors of a mote i, that is

$$\forall i \in V, \quad N(i) = \{j \in V| \quad ij \in E\}$$

and

$$\forall A \subset V, \quad N(A) = \bigcup_{i \in A} N(i)$$

Let $G_{2+}(V, E_{2+})$ define the graph where

$$\forall i, j \in V^2, \quad ij \in E_{2+} \text{ iff } j \in N(N(i)) \backslash i$$

The graph G_{2+} is the graph of all motes which are at most 2-distant between them. Let $S(.)$ denote the mapping which maps a vertex $v \in V$ to the number of birds counted locally. Let $\mathcal{C}$ be the set of all independent set in graph G_{2+}. Our estimation is the sum of birds counted locally for each motes derived from the maximum weighted independent set of G_{2+}, i.e

$$\max_{c \in \mathcal{C}} \sum_{v \in c} S(v)$$

Lemma 5.1. *G_{2+} is a chordal graph.*

Proof. Proof Remark that in G_{2+}, all vertices are simplicial[4]. Thus, there exists a perfect elimination ordering on its vertices and de facto G_{2+} is chordal. □

[4] Vertices for which neighbors induce a clique in the graph.

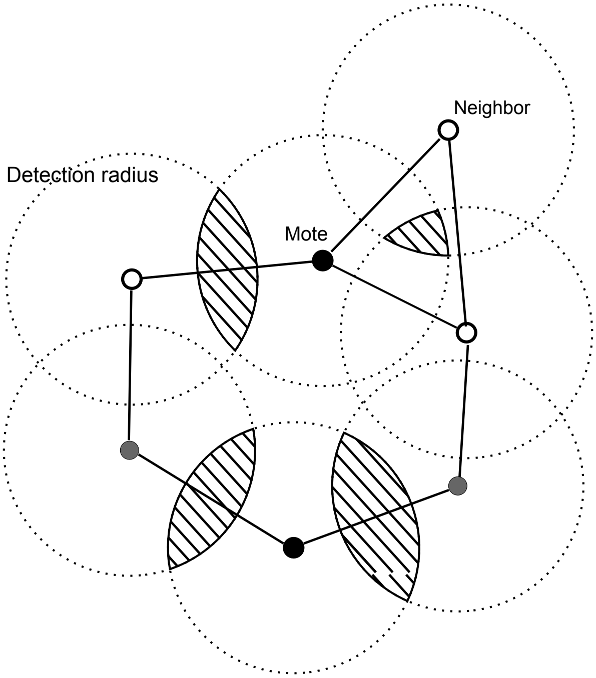

Fig. 4. Example of underlying unit disk graph in local and global detection

Chordal graphs are graph for which vertices do not induce cycles without chord of size more or equal to four. They are perfect graphs and well discussed in (Golumbic, 1980). It is also well known that finding a maximum weighted independent set in chordal graph is linear (Leung, 1984). Thus, our later algorithm finds its estimation of the number of birds in linear time given G_{2+}.

Let's see why and how our algorithm is not so sensible to noise and encompasses non circular detection area. One of the most interesting features of swarm computing (Blum & Merkle, 2008) is that nodes (swarm entities) create mechanisms which tend to be resilient to disruption and failure. Similarly, our last counting technique leverages the swarm intelligence since motes collaborates each other to derive their local count. The more the motes are, better the estimate is. There are two cases where inconsistencies could appear :

1. Motes can have a different status from what it would be. For example, a mote could stay in a passive state while it would have heard "a bird song". However, neighbor motes tend to negate this last effect. Conversely, motes could "wake up" while no birds have sung. This latter case is somewhat less frequent and is easier to correct since this mote could be a one-vertex connected component in the underlying graph, fact which is prone to be an erratic behavior of the mote.
2. Objects can occlude bird songs, that is detection area is no more circular. In that case, the occluded motes would stay in a passive state. Fortunately, the swarm could correct this drawback by multiplicity : other closer motes could hear the birds too.

Therefore, note that the layout of the motes is somewhat important and a simple way to tackle the occlusion problem is to rise the density of the motes on the monitored environment. It is even possible to only increase the number of motes where occlusion problems could occur.

The next section is dedicated to experiments which prove our algorithm efficiency, even in the presence of noise.

6. Experiments

6.1 Context

Endangered species receive attention from the scientific community since their disappearance would lead to irreplaceable losses. To help these species to survive, their habitat is protected by laws of environmental protection. Sometimes, this protection is not sufficient since their habitat evolution is the main cause of their vanishing. In order to save them, they must be transferred elsewhere. Obviously, the new habitat has to be similar to the previous one to minimally disrupt the equilibrium of the wildlife. To model a habitat, several parameters have to be fixed by an expert. This study precedes the MOM project for which wireless sensor networks have to be used to monitor an endangered specie. So these simulations are the first steps to the deployment of WSNs over the Caravelle location in Martinique (a French Caribbean island). Indeed, birds called "White-breasted Thrasher" are a specie which is only known to be in in the Caravelle. They are considered endangered since specialists think that only fifty of them are still alive there.

6.2 Testbed

Environment

For the need of the simulations, we wrote a tool which aims at generating the data necessary to run our counting heuristics described previously. Our test environment comprises :

- an Intel Core 2 Duo E6750 2.67 GHZ,
- 4 Go RAM,
- Windows Vista 64 bits for Operating System,
- and the JDK 1.6 Update 10 (x64) since our tool is written in java.

Parameters

Simulations parameters were calibrated to be the closest to our tested area. The dimension of our habitat is about 1000m×1000m. Birds can fly at 2 meters per second, stay at place, take random directions with uniform probability. They sing with some probability fixed empirically. This latter parameter is fixed at 0.2 for each sample record (detection array). Finally, motes are placed randomly on our area.

6.3 Performance Evaluation

Figure 5 shows our three different algorithms estimation for counting fifty living birds in the habitat. *Algo1* stands for the algorithm which only rely on the underlying UDG. *Longest* is the algorithm which alters the longest edges of "bad triangles". *Swarm* is the algorithm presented in section 5. *True* is the number of birds which really sang. Each test has been driven 50 times and the mean of the estimations was taken as the final result. Error deviation is shown for

Motes	Algo1	Longest	Swarm	True
100.00	10.10	10.72	9.98	17.58
200.00	13.86	16.56	12.62	17.38
300.00	15.16	18.64	12.92	18.02
400.00	15.70	20.46	13.84	17.58
500.00	16.76	22.56	15.04	17.36
600.00	17.28	24.12	15.80	17.80
700.00	17.88	25.74	16.08	17.66
800.00	18.38	27.74	17.04	17.64
900.00	18.40	29.50	17.60	17.86
1000.00	18.30	31.22	17.98	17.80

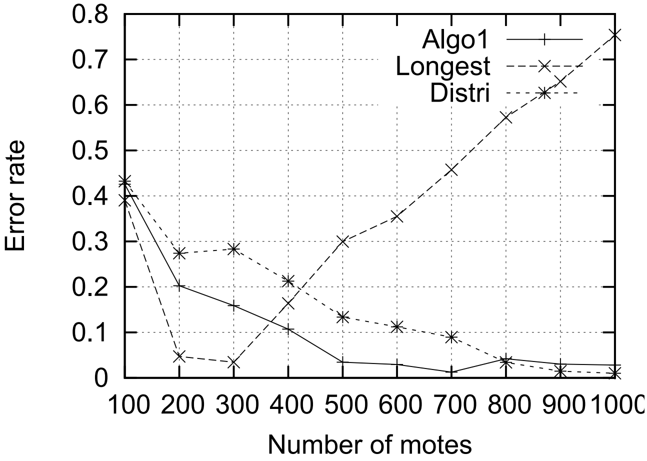

Fig. 5. Number of Birds found by the heuristics and error deviation for 50 Birds in the habitat

each algorithm on the graph near the tables and gives an idea of how the algorithms perform along the parameters.

Experiments show that algorithm *Algo1* performs nearly as well as algorithm *Swarm* whenever the number of motes is high. However, *Algo1* tends to over-count the birds. The outputted number of birds depends on the manner the MDS are removed. Let's sketch a brief example on figure 6. There exists three MDS at first step in each configuration. Grayed areas represent the MDS removed on each configuration at each step. Configuration 1 leads to two grayed area whereas configuration 2 leads to three grayed area. That is how two birds can be counted as three. To reduce this drawback, we could run several times the counting process which would remove the MDS randomly and then take the minimum number of birds over these countings. However, such a scheme does not guarantee that we will not over-count and further, will highly rise the execution time of the whole process.

To validate our schemes, noise is added as a parameter in our simulator. We assume that 20% of the motes malfunction. Table 1 shows the percentage error for *Algo1* and *Swarm* in the presence of noise.

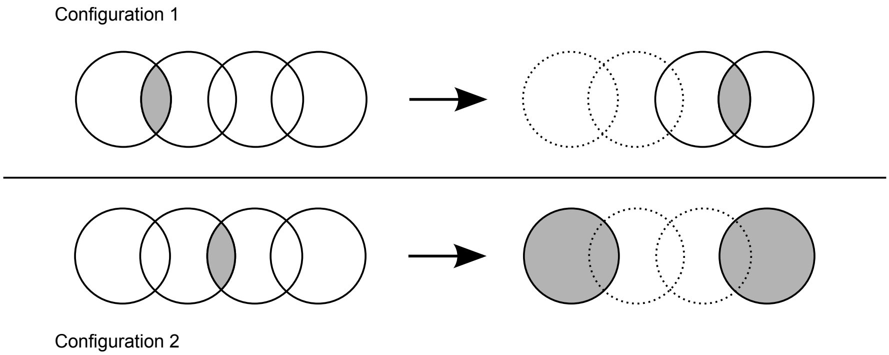

Fig. 6. Example of bad counting

Table 1. Relative error for the counting algorithms for 50 birds

Motes	Algo1	Swarm	Algo1 with noise (%)	Swarm with noise(%)
100	45,08	46,33	50,51	50,85
200	27,24	33,83	32,69	37,34
300	15,27	25,79	20,59	28,62
400	8,85	19,31	13,33	23,22
500	9,32	19,77	13,75	23,86
600	3,31	12,69	6,86	17,49
700	1,93	4,54	1,36	9,53
800	4,41	3,72	0,93	7,67
900	4,30	2,49	2,04	6,23
1000	7,11	1,49	4,47	1,03

Clearly, the algorithm based on the swarm counting protocol seems more sensitive to noise than its counterpart. Note that without noise, algorithm *Algo1* over-counts the number of birds. Therefore, in presence of noise, the approximate of the number of birds tends to be more precise. Conversely, algorithm *Swarm* already undercounts the number of birds originally. So, noise degrades even more the approximate of the number of birds which generally leads to a worse counting.

Finally, we decided to fix the number of motes which will be used on the Caravelle habitat to 1000 and vary the number of birds in our simulator to confirm our estimation. Results are shown on figure 7.
Algorithm *Longest* suffers the same drawback seen in figure 5 when the number of motes is high, so much that estimation are too high. This results from the fact that altering triangles tends to create much more cliques to remove.
Algorithm *Swarm* gives slightly better estimations in configuration using a high number of motes. However, its efficiency lowers whenever less motes are used. Indeed, the lesser the motes you have, the lesser you cover the habitat. Furthermore, our schemes rely on a high number of motes to better estimate the singing birds except algorithm *Longest* which could be used to estimate the songbirds whenever the number of motes are low.

Birds	Algo1	Longest	Swarm	True
10.00	5.94	10.20	5.44	5.34
20.00	9.08	16.08	8.76	8.90
30.00	12.76	20.58	12.20	12.28
40.00	15.90	25.48	15.06	14.76
50.00	18.30	31.22	17.98	17.80
60.00	21.64	34.76	20.62	20.28
70.00	23.92	37.74	22.56	23.08

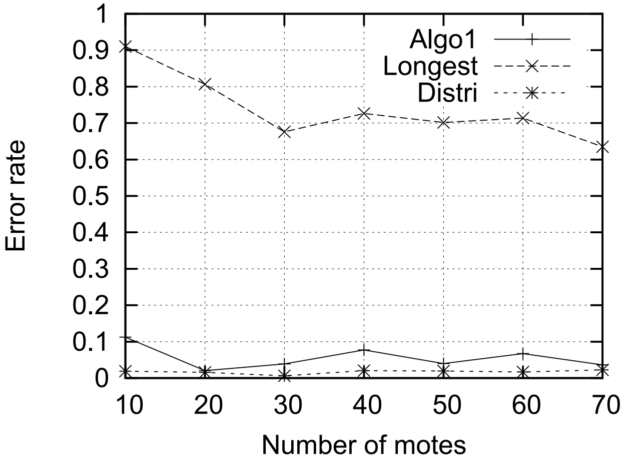

Fig. 7. Number of Birds found by the heuristics and error deviation for 1000 motes in the habitat

These results suggest to design an hybrid algorithm which will switch along determined thresholds. However, remind that the data was generated from simulations and these thresholds could be different from what our experimentations outputted. In our case, using the *Swarm* algorithm for counting the birds seems to be so far the best solution to apply in the *Caravelle* since more motes give accurate estimations.

7. Conclusion

Endangered species are a known problem that drew attention from the community these last years. Habitat monitoring with wireless sensors networks could lead to several improvements in the way to tackle the problem of the survival of these species. We have proposed a first technique to estimate the number of birds using wireless microphone motes scattered in an habitat. Our method derives from the motes layout a unit disk graph and removes continuously maximum cliques to count the number of birds. A limitation of this technique could be the maximum clique problem but simulations have shown that estimations are still suitable if unit disk graphs are used to represent the motes network. We have also proposed a linear algorithm to estimate the singing birds in the habitat and have shown that it is as much as efficient (quality) as our first one. This scheme can be fully distributed on a suitable wireless sensors network. Such a distributed scheme would deny the need of a powerful base station since the counting process would totally shift from the base to the motes.

Counting singing birds is a first step in our habitat monitoring project and surely is not sufficient to identify specificities of the monitored specie. One major goal of habitat monitoring is the reintroduction of the specie in another environment which will share the same characteristics. We intend to work in this way by monitoring several parameters of interests in an environment to model it and compare it with another ones.

8. References

Bell, R. E. (1964). A Sound Triangulation Method for Counting Barred Owls, *The Wilson Bulletin*, The Wilson Ornithological Society.

Blum, C. & Merkle, D. (eds) (2008). *Swarm Intelligence: Introduction and Applications*, Natural Computing Series, Springer.
URL: *http://dx.doi.org/10.1007/978-3-540-74089-6*

Bomze, I. M., Budinich, M., Pardalos, P. M. & Pelillo, M. (1999). The Maximum Clique Problem, *Research Report CS-99-1*, Dipartimento di Informatica, Univerità Ca' Foscari di Venezia.

Breu, H. (1996). Algorithmic aspects of constrained unit disk graphs, *Technical Report TR-96-15*, Department of Computer Science, University of British Columbia. Tue, 22 Jul 1997 22:20:10 GMT.
URL: *ftp://ftp.cs.ubc.ca/pub/local/techreports/1996/TR-96-15.ps.gz*

Cai, J., Ee, D., Pham, B., Roe, P. & Zhang, J. (2007). Sensor network for the monitoring of ecosystem: Bird species recognition, *Intelligent Sensors, Sensor Networks and Information*, Queensland Univ. of Technol., Brisbane, pp. 293–298.

Ceroi, S. (2002). The clique number of unit quasi-disk graphs, *Rapport*.
URL: *http://hal.inria.fr/inria-00072169/en/; http://hal.ccsd.cnrs.fr/docs/00/07/21/69/PDF/RR-4419.pdf*

Chen, J. C., Hudson, R. E. & Yao, K. (2001a). A maximum-likelihood parametric approach to source localizations, *Proceedings of the Acoustics, Speech, and Signal Processing*, IEEE Computer Society, Washington, DC, USA, pp. 3013–3016.

Chen, J. C., Hudson, R. E. & Yao, K. (2001b). Joint maximum-likelihood source localization and unknown sensor location estimation for near-field wideband signals, *in* F. T. Luk (ed.), *Advanced Signal Processing Algorithms, Architectures, and Implementations XI*, Vol. 4474, SPIE, pp. 521–532.
URL: *http://link.aip.org/link/?PSI/4474/521/1*

Clark, B. N., Colbourn, C. J. & Johnson, D. S. (1990). Unit disk graphs, *Discrete Math.* **86**(1-3): 165–177.

Fagerlund, S. (2007). Bird species recognition using support vector machines, *EURASIP J. Appl. Signal Process.* **2007**(1): 64–64.

Golumbic, M. C. (1980). *Algorithmic Graph Theory and Perfect Graphs*, Academic Press.

Krim, H. & Viberg, M. (1996). Two decades of array signal processing research — the parametric approach, *IEEE Signal Processing Magazine*, Vol. 13(3).

Kuhn, F., Wattenhofer, R. & Zollinger, A. (2008). Ad hoc networks beyond unit disk graphs, *Wireless Networks* **14**(5): 715–729.
URL: *http://dx.doi.org/10.1007/s11276-007-0045-6*

Leung, J. Y. T. (1984). Fast algorithms for generating all maximal independent sets of interval, circular-arc and chordal graphs, *ALGORITHMS: Journal of Algorithms* **5**.

Lévy, C., Linarès, G. & Bonastre, J.-F. (2006). Gmm-based acoustic modeling for embedded speech recognition, *In International Conference on Speech Communication and Technology*.

Rabiner, L. R. & Wilpon, J. G. (1979). Considerations in applying clustering techniques to speaker-independent word recognition, *The Journal of the Acoustical Society of America* **66**(3): 663–673.
URL: *http://link.aip.org/link/?JAS/66/663/1*

Raghavan, V. & Spinrad, J. (2003). Robust algorithms for restricted domains, *ALGORITHMS: Journal of Algorithms* **48**.

Reichling, M. (1988). On the detection of a common intersection of k convex objects in the plane, *Information Processing Letters* **29**(1): 25–29.

Schmidt, R. (1986). Multiple emitter location and signal parameter estimation, *IEEE Transactions on Antennas and Propagation* **34**(3): 276–280.

Smith., J. & Abel, J. (1987). Closed-form least-squares source location estimation from range-difference measurements, *IEEE Transactions on Acoustics, Speech and Signal Processing* **35**(12): 1661–1669.

Wireless Sensor Networks: Application-Centric Design
Edited by Yen Kheng Tan

ISBN 978-953-307-321-7
Hard cover, 492 pages
Publisher InTech
Published online 14, December, 2010
Published in print edition December, 2010

Over the past decade, there has been a prolific increase in the research, development and commercialisation of Wireless Sensor Networks (WSNs) and their associated technologies. WSNs have found application in a vast range of different domains, scenarios and disciplines. These have included healthcare, defence and security, environmental monitoring and building/structural health monitoring. However, as a result of the broad array of pertinent applications, WSN researchers have also realised the application specificity of the domain; it is incredibly difficult, if not impossible, to find an application-independent solution to most WSN problems. Hence, research into WSNs dictates the adoption of an application-centric design process. This book is not intended to be a comprehensive review of all WSN applications and deployments to date. Instead, it is a collection of state-of-the-art research papers discussing current applications and deployment experiences, but also the communication and data processing technologies that are fundamental in further developing solutions to applications. Whilst a common foundation is retained through all chapters, this book contains a broad array of often differing interpretations, configurations and limitations of WSNs, and this highlights the diversity of this ever-changing research area. The chapters have been categorised into three distinct sections: applications and case studies, communication and networking, and information and data processing. The readership of this book is intended to be postgraduate/postdoctoral researchers and professional engineers, though some of the chapters may be of relevance to interested masterâ€™s level students.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Harry Gros-desormeaux, Philippe Hunel and Nicolas Vidot (2010). Wildlife Assessment Using Wireless Sensor Networks, Wireless Sensor Networks: Application-Centric Design, Yen Kheng Tan (Ed.), ISBN: 978-953-307-321-7, InTech, Available from: http://www.intechopen.com/books/wireless-sensor-networks-application-centric-design/wildlife-assessment-using-wireless-sensor-networks

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820

Fax: +385 (51) 686 166
www.intechopen.com

Fax: +86-21-62489821