

Low-Complexity Audio Signal Processing for Localization in Indoor Scenarios

Marco Martalò and Gianluigi Ferrari

1 Introduction

During the last years, wireless sensor networks (WSNs) have received significant attention from the research community, as one of the emerging technologies for the new millennium. A WSN is composed by many (e.g., hundreds) devices with limited processing and communication capabilities. Therefore, energy saving is one of the major issues and information processing has to be performed with low complexity. Possible applications for WSNs are surveillance, environmental monitoring, flow control, etc, and it may be possible to work in indoor scenarios [1]. The application of interest in this paper is the localization of a person or an object in indoor scenarios [2], but our approach is also suitable for outdoor scenarios. We will refer to the person (or object) to be localized as the *entity*. In these scenarios, it is realistic to assume that the nodes are *not* equipped with global positioning system (GPS) devices and, therefore, other techniques are needed to perform efficient localization.

Most of the works about localization in sensor networks are based on the assumption that the entities are equipped with devices which radio-communicate with some reference nodes (denoted as *anchors*). The positions of the anchors are supposed to be known and the position of the entity of interest is inferred by “combining” the information available at each anchor (e.g., by triangulation). In the literature, several techniques, based on different methods, have been proposed to obtain a sufficiently low estimation error [3, 4]. The computational complexity of these algorithms is a crucial issue for WSN-based applications. In [5], the authors propose a sub-optimal hierarchical algorithm, which solves the localization problem without resorting to the optimum maximum likelihood (ML) technique, whose computational complexity becomes too high to be of any practical interest. In [6], an adaptive approach to localization problems, obtained by solving a sequence of very small optimization subproblems, is considered.

M. Martalò (✉) and G. Ferrari
WASN Lab, Department of Information Engineering, University of Parma, Parma, Italy
e-mail: martalo@tlc.unipr.it; gianluigi.ferrari@unipr.it

The problem of locating a source of speech has been widely studied in the field of sound source localization (SSL) using multiple input multiple output (MIMO) signal processing [7]. In particular, several analytical frameworks have been proposed for the estimation of the time difference of arrival (TDOA). Most of the methods are based on measuring the crosscorrelation (in time or frequency domains) between the output at different receivers [8, 9]. An interesting approach is that proposed in [10], where the authors derive a unified ML framework for sound source localization and beamforming for distributed meeting applications, taking into account both reverberation and environmental noise.

In this chapter, we will assume that the anchors are equipped with microphones (which have, typically, a low cost) and use the information collected by them to localize, through collaborative SSL-based signal processing, the entity of interest. This is reasonable in scenarios where the entity to be localized may not be equipped with these devices, e.g., when the entity is an enemy to be located in a battlefield. In such cases, it is thus necessary to use other methods to localize the entity, e.g., by employing other types of sensors, such as accelerometers, microphones, etc. Although SSL techniques are well established (especially for distributed meeting applications), they are mainly based on the computation of crosscorrelation and the use of ML estimators, which are computationally onerous. *In this chapter*, we derive SSL techniques which employ very limited computational complexity, trying to obtain the minimum penalty in terms of position estimation error. In particular, we present results based on a novel localization algorithm which, by considering the powers of the audio signals received at the microphones, determines the position of the entity. We first deal with one-dimensional scenarios, i.e., scenarios where the audio source moves along a straight line, deriving both *centralized* and *decentralized* localization algorithms, based on the solution of simple systems of equations. Then, we extend our approach to consider more realistic two-dimensional scenarios, where the anchor nodes are placed at the corners of a square grid.

2 One-Dimensional Scenarios

2.1 Statement of the Problem

Suppose that the entity to be localized is moving on a straight line (x -axis) and there are N anchors (microphones), denoted as $[m_0, \dots, m_{N-1}]$, equally spaced at positions $[x_0, \dots, x_{N-1}]$, where

$$x_j = x_0 + D \cdot j \quad j = 1, \dots, N - 1$$

and D is the constant distance between two consecutive anchors. Without loss of generality, suppose that $x_0 = 0$. This approach can be easily extended to scenarios with nonequally spaced sensors.

The audio power received at the i th node ($i = 0, 1, \dots, N - 1$) can be expressed as

$$P_i = k \frac{P_s}{(d_i)^\beta} \quad (1)$$

where k is a suitable parameter which depends on the audio propagation characteristics (dimension: $[\text{cm}^\beta]$), β is the pathloss decay exponent (adimensional), d_i is the distance of the audio source from m_i (dimension: $[\text{cm}]$), and P_s is the maximum emitted power by the sound source (dimension: $[\text{W}]$). Obviously, expression (1) can be applied provided that $d_i > d_{\text{crit}}$, where d_{crit} is a critical distance beyond which equation (1) holds. Although a more detailed statistical description of the model may be needed, results on acoustic emission of human head confirm this type of model [11]. Note that β depends on the type of audio source. If, for instance, the sound is emitted by an object (e.g., a motor), it is reasonable to assume that the sound may approximately have the same propagation characteristics in all directions (positive and negative directions on the x -axis). If, instead, the sound is emitted by a person, the sound will decay slowly in front of the face, whereas it will decay faster on the opposite direction. For the ease of simplicity, we will assume that $\beta = \beta_{\text{forward}} = 2$ in the “forward” direction, whereas $\beta = \beta_{\text{backward}} = 4$ in the “backward” direction. We remark that in an homogenous scenario the propagation exponent β should be the same in all directions. For the purpose of analysis and without leading the generality of our framework, we suppose that β depends on the propagation direction. A different propagation modeling would simply require to change a few equations. However, the value of β changes from person to person and a more accurate description may be needed [11]. As will be shown later, an accurate characterization of k is not crucial in our analytical framework, since the same value of k is considered for all sensors.

The problem consists in locating the audio source on the basis of the N audio power received at the anchors. Our goal is to derive an efficient cooperative processing algorithm, with low computational complexity, for the localization of the entity. In Sect. 2.2, we will derive a centralized algorithm in the considered one-dimensional scenarios, whereas in Sect. 2.3, distributed localization algorithms will be proposed.

2.2 Centralized Localization Algorithm

In a scenario with *omnidirectional* audio source emission (e.g., a motor), the parameter β is the same in all directions. Therefore, by identifying the two nodes which receive the highest audio powers, one can determine the position of the entity. For instance, consider the scenario depicted in Fig. 1, where the source is between m_{i-1} and m_i . The powers P_{i-1} and P_i received at m_{i-1} and m_i , respectively, will be the highest ones. The distance x_{crit}^* of the audio source from m_{i-1} , i.e., $x_{\text{crit}}^* = x^* - (i - 1)D$, can be directly obtained, once the powers at the $(i - 1)$ th and i th sensors are collected.

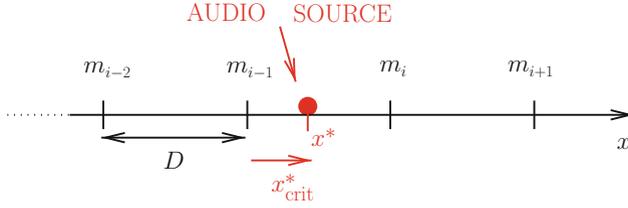


Fig. 1 Reference one-dimensional scenario

In the presence of *directive* audio sources (e.g., a human speaker), the above approach cannot be applied, since the value of β changes according to the direction in which the sound is emitted. For example, if the entity is between m_{i-1} and m_i and speaks “forward” (i.e., in the positive direction of the x -axis), even if $P_i > P_{i-1}$, it might happen that the entity is closer to m_{i-1} than to m_i . Vice-versa, if the entity is speaking “backward” and $P_i > P_{i-1}$, then for sure the entity is closer to m_i . It might even happen (depending on the value of D and the values of $\beta_{forward}$ and $\beta_{backward}$) that if the entity is on the left of m_{i-1} (i.e., between m_{i-2} and m_{i-1}) and speaks forward, then the powers perceived at m_{i-1} , m_i , and m_{i+1} are the highest ones.

As can be understood from the illustrative examples in the previous paragraph, a generalized approach needs to be considered when the “direction” of the sound (forward or backward) has also to be determined. In this case, one needs to consider at least three nodes. Suppose that the highest received powers are those perceived at the anchors m_{i-1} , m_i , and m_{i+1} (the order is not relevant). Then, only one of the following exclusive situations can happen:

1. The entity speaks forward and is between m_{i-2} and m_{i-1}
2. The entity speaks forward and is between m_{i-1} and m_i
3. The entity speaks backward and is between m_{i-1} and m_i
4. The entity speaks forward and is between m_i and m_{i+1}
5. The entity speaks backward and is between m_i and m_{i+1}
6. The entity speaks backward and is between m_{i+1} and m_{i+2}

Each of the above conditions is associated with a specific system. For example, in the first case, assuming, as mentioned in Sect. 1, that $\beta_{forward} = 2$, the following system admits a unique solution $x^*_{crit} < 0$:

$$\begin{cases} P_{i-1} = k \frac{P_s}{(x_{crit})^2} \\ P_i = k \frac{P_s}{(D - x_{crit})^2} \\ P_{i+1} = k \frac{P_s}{(2D - x_{crit})^2} \end{cases}$$

In the second case, assuming $\beta_{backward} = 4$, the following system admits a unique solution $x^*_{crit} > 0$:

$$\begin{cases} P_{i-1} = k \frac{P_s}{(x_{\text{crit}})^4} \\ P_i = k \frac{P_s}{(D - x_{\text{crit}})^2} \\ P_{i+1} = k \frac{P_s}{(2D - x_{\text{crit}})^2} \end{cases}$$

In general, there will be only one system (out of the six possible ones) which will admit an acceptable solution ($-D < x_{\text{crit}}^* < 2D$). Therefore, the audio source location and direction can be univocally determined by finding such a system.

2.3 Distributed Localization Algorithms

Although the analytical framework described above is very simple to be implemented, it requires a global network knowledge, since the three highest received powers (among all the N powers received at the anchors) are used to determine the audio source position and direction of emission. In practical networks, a centralized solution may not be feasible, since extra nodes with higher computational resources may be required. Therefore, it is of interest to derive distributed algorithms, where data are gathered and disseminated with the smallest possible number of inter-anchor communications. In the literature, several distributed algorithms have been proposed, based especially on the use of machine learning techniques [12, 13]. In the following, we derive two possible distributed strategies. The common feature of these strategies is that only the three nodes with the highest received powers are involved in the possible systems described at the end of Sect. 2.2. In other words, the location of the entity is determined from the data perceived by three anchors and, then, disseminated to all other nodes – this might be of interest for tracking operations.

The first proposed protocol can be described as follows:

- A packet with the information about the three nodes with the highest powers is created at node m_0 and propagated along the x -axis. At the first transmission act, node m_0 only stores its received power.
- If an anchor receiving the packet has a measured power higher than any of the three stored in the packet, it modifies the packet by discarding the lowest power and introducing its own.
- If a node has a received power lower than the three powers collected in the packet, data gathering stops, since it is not possible (according to the propagation model (1)) to find forward an anchor with higher received power.
- Once the nodes with the highest received powers are identified, the one-dimensional localization algorithm described in Sect. 2.2 is carried out at these nodes.
- Finally, the estimated position and direction are disseminated through the network to all other nodes.

While it is possible to show that $O(N/2)$ interanchor communications are needed, on average, during the data gathering phase, one should note that in the worst case scenario the number of transmissions is $O(N)$. Therefore, one may design a more efficient gathering algorithm which starts from the middle of the anchor sequence, instead of one edge, of the network. This protocol can be described as follows.

- A packet with the information about the three nodes with the highest powers is created at node $m_{\lfloor N/2 \rfloor}$ and it is propagated along the two directions of the x -axis. As in the previous case, anchor $m_{\lfloor N/2 \rfloor}$ stores only its power.
- For each direction, if a node receiving the packet (with already three stored values) has a higher measured power, it modifies the packet by discarding the lowest power and introducing its own; the gathering phase stops when the three highest are collected (in each direction).
- The three nodes with the overall highest received powers are determined according to the information in the packets collected above.
- Once the nodes with the highest received powers are identified, the one-dimensional localization algorithm described in Sect. 2.2 is carried out at these nodes.
- Finally, the estimated position and direction are disseminated through the network to all other nodes.

In Fig. 2, the average number of interanchor communications \bar{n}_{tx} is shown, as a function of the number of nodes, for the two distributed protocols described above. The interanchor distance is set to $D = 50$ cm. To this regard an ad hoc simulator, written in Matlab [14], has been created. The average is computed by simulating different (independent) positions and directions, computing the number of communication steps for each run and, finally, averaging. As expected, \bar{n}_{tx} is an increasing function of the number of anchors, since, on average, one may need more steps before reaching the nodes with the highest received powers. However, one can observe that the second protocol is more efficient, since it requires a smaller number of communication steps before completing the localization process. This is due to

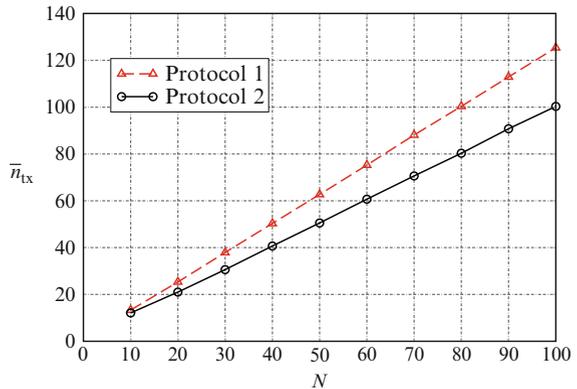


Fig. 2 Average number of communications \bar{n}_{tx} , as a function of the number of nodes, for the two distributed protocols described above. The interanchor distance is set to $D = 50$ cm

the fact that the first protocol is “unbalanced,” i.e., it may happen that the audio source is close to m_{N-1} . The second protocol, instead, is more “balanced,” since it never happens that the audio source is N steps away from the anchor ($m_{\lfloor N/2 \rfloor}$) which initializes the data gathering phase.

3 Two-Dimensional Scenarios

3.1 Statement of the Problem

Suppose that the entity to be localized is moving on a square area of side D (e.g., a room), and there are four microphones, denoted as $[m_1, m_2, m_3, m_4]$, equally spaced at the corners of the square area. Without loss of generality, suppose that the origin of the axes is at the center of the square area. Therefore, the anchors are placed at $(\pm D/2, \pm D/2)$.

The audio power received at the i th anchor ($i = 1, 2, 3, 4$) can be expressed as in (1), where d_i is now the euclidean distance of the audio source, located at (x^*, y^*) , from m_i , i.e.,

$$d_i = \sqrt{|x^* - x_i|^2 + |y^* - y_i|^2}.$$

As in the one-dimensional case, in this case as well β depends on the type of audio source and sound emission. For ease of simplicity, we assume that the person can speak only along one of the axes. In this case, we assume that the decay factor in a frontal region of span angle θ is equal to $\beta = \beta_{\text{forward}} = 2$, otherwise it is $\beta = \beta_{\text{backward}} = 4$. In a scenario with *omnidirectional* audio source emission (e.g., a motor), the parameter β is the same in all directions, i.e., $\theta = 2\pi$. Therefore, by identifying the audio power distribution among the nodes, one can determine the position of the entity by solving (1) for each anchor node. A pictorial description of the scenario is given in Fig. 3.

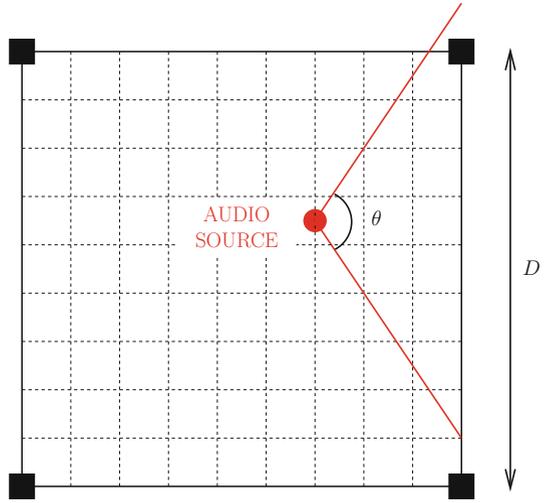
The problem consists in locating the audio source on the basis of the four audio powers received at the anchors and the knowledge of the sound emission characteristics, i.e., θ .

3.2 Centralized Localization Algorithm

Without loss of generality, we focus on the case when the entity speaks toward the side delimited by m_1 and m_4 , i.e., the “eastern side” of the area. However, similar considerations can be carried out for the other three sides.

In order to determine the position of the entity, the value of β in (1), at all four anchors, has to be known. One should note that, if the entity is close to the center of

Fig. 3 Reference two-dimensional scenario



the side, none of the microphones will observe a decay factor equal to 2, since none of the microphones is spanned by the forward emission lobe of the entity. On the other hand, if the entity is close to the other side but still in the middle, m_1 and m_4 will observe a value of β equal to 2, whereas m_2 and m_3 will observe a value of β equal to 4. Moreover, only m_1 (m_4 , respectively) will observe $\beta = 2$ if the entity is in the upper (bottom, respectively) part of the area. Therefore, one has to simply compute the equations of the straight lines that divide these four sectors. After a few geometrical considerations, denoting as β_i the decay factor at anchor i , it is possible to verify that

$$\left\{ \begin{array}{ll} \beta_i = 4 \quad \forall i & \text{if } x > x_{\text{crit}} \text{ and } |y| < m|x| + q \\ \beta_1 = \beta_4 = 2, \beta_2 = \beta_3 = 4 & \text{if } x \leq x_{\text{crit}}, y < -(mx + q), \text{ and } y > mx + q \\ \beta_1 = 2, \beta_2 = \beta_3 = \beta_4 = 4 & \text{if } y > 0, x < (y - q)/m, \text{ and } x > -(y + q)/m \\ \beta_4 = 1, \beta_1 = \beta_2 = \beta_3 = 4 & \text{otherwise} \end{array} \right.$$

where $x_{\text{crit}} = (D/2) \tan((\pi - \theta)/2)$, $m = \tan(\theta/2)$, and $q = -x_{\text{crit}} \cdot m$.

If one applies similar considerations to the other three sides, it is possible to identify a set of 16 systems (four for each possible direction). As one can see, the computational complexity has increased rapidly. In fact, we have increased only one dimension of the problem, increasing the number of systems to be solved from 6 to 16 systems. However, the complexity remains acceptable, since solving 16 systems has still lower complexity than that of an ML-based algorithm.

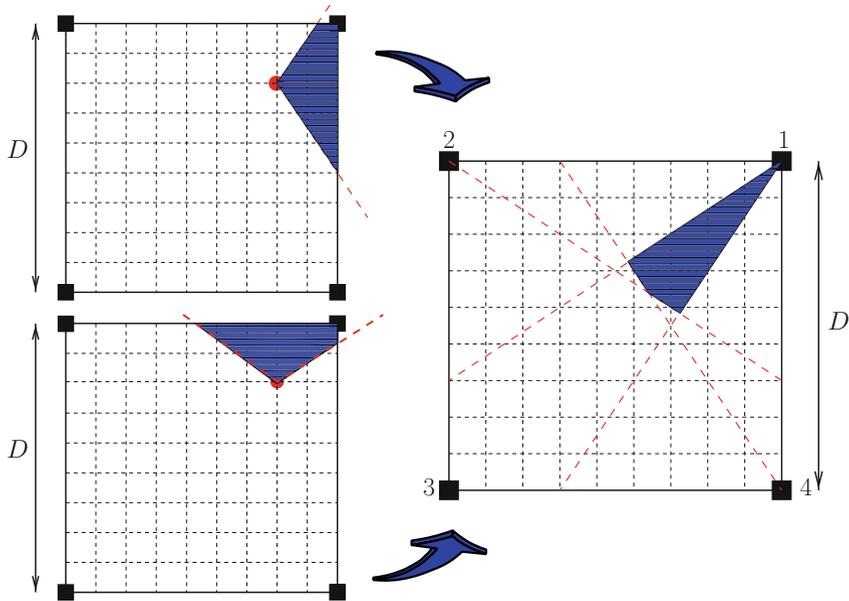


Fig. 4 Ambiguity zone for the anchor m_1 in a two-dimensional scenario

4 Results and Discussion

As previously anticipated, we have performed simulations using Matlab, in order to determine the effectiveness of the proposed approaches. In particular, we have compute the average errors in estimating the position and direction, denoted as ε_p and ε_d , respectively, by averaging over 1,000 independent trials the differences between the true position and direction and their estimates. During each trial, different positions and directions of the entity are randomly generated. The distance between consecutive anchors is set, in all cases, to $D = 50$ cm.

For one-dimensional scenarios, we found that $\varepsilon_p^{(1-dim)} = \varepsilon_d^{(1-dim)} = 0$, thus confirming the uniqueness of the solution of the 6 systems. In two-dimensional scenarios, we still found that $\varepsilon_p^{(2-dim)} = 0$, but $\varepsilon_d^{(2-dim)} \simeq 0.1$. In other words, the position is still correctly determined in all cases, but in 10% of the cases, the estimated direction is erroneous. A more detailed analysis has shown that this is due to the fact that there exists *two* possible systems with the same values of β_i and d_i ($i = 1, 2, 3, 4$). In this case, the position can be correctly estimated, but the direction is ambiguous. An example of the ambiguity zone for the anchor m_1 is shown in Fig. 4. In this region, there could be uncertainty between the emission directions toward the eastern or northern side. A proper strategy to solve this ambiguity still remains an open problem.

Acknowledgments We would like to thank Sandro Mattiacci, Claudio Malavenda, Luca Di Donato, and Paolo Proietti (Elsag Datamat S.p.A, Rome, Italy) for useful discussions on localization issues and audio signal processing. This work has been supported by a SPINNER 2013 fellowship.

5 Concluding Remarks

In this chapter, we have proposed a novel approach to perform low-complexity localization based on audio signal processing. A set of “anchors,” which perceives the sound intensity (through audio sensors) emitted by an “entity” and collaborate together, estimate (a) the position of the entity and (b) the direction of sound emission. We have derived a framework for both one and two-dimensional scenarios, also showing possible distributed approaches in the one-dimensional case. Since ideal sound propagation conditions (i.e., no noise) have been assumed, the future work will be devoted to the derivation of proper techniques to counter-act the presence of acquisition and communication noises.

References

1. Akyildiz I, Su W, Sankarasubramaniam Y, Cayirci E (2002) A survey on sensor networks. *IEEE Commun Mag* 40(8):102–114
2. Bachrach J, Taylor C (2005) Localization in sensor networks. In: Stojmenović I (ed) *Handbook of sensor networks: algorithms and architectures*. Wiley, New York
3. Dricot J-M, Bontempi G, Doncker PD (2010) Static and dynamic localization techniques for wireless sensor networks. In: Ferrari G (ed) *Sensor networks: where theory meets practice*, Springer, pp 249–281
4. Savarese C, Rabaey J-M, Beutel J (2001) Locationing in distributed ad-hoc wireless networks, In: *Proceedings of the IEEE international conference acoustics, speech, and signal processing (ICASSP)*, vol 4. Salt Lake City, UT, May 2001, pp 2037–2040
5. Dardari D, Conti A (2004) A sub-optimal hierarchical maximum likelihood algorithm for collaborative localization in ad-hoc networks. In: *Proceedings of the IEEE sensor and ad hoc communications and networks (SECON)*, Santa Clara, CA, October 2004, pp 425–429
6. Carter M-W, Jin H-H, Saunders M-A, Ye J (2006) SpaseLoc: an adaptive subproblem algorithm for scalable wireless sensor network localization. *SIAM J Optim* 17(4):1102–1108
7. Huang Y, Benesty J, Chen J (2006) *Acoustic MIMO signal processing*. Springer, Heidelberg
8. Brandstein M-S, Adcock J-E, Silverman H-F (1997) A closed-form location estimator for use with room environment microphone arrays. *IEEE Trans Acoust Speech Signal Process* 5(1):45–50
9. Brandstein M-S, Adcock J-E, Silverman H-F (1995) A practical time-delay estimator for localizing speech sources with a microphone array. *Comput Speech Lang* 9(2):153–169
10. Zhang C, Florencio D, Ba D-E, Zhang Z (2008) Maximum likelihood sound source localization and beamforming for directional microphone arrays in distributed meetings. *IEEE Trans Multimed* 10(30):238–248
11. Dunn H-K, Farnsworth D-W (1939) Exploration of pressure field around the human head during speech. *J Acoust Soc Am* 10(1):184–199
12. Roos T, Myllymaki P, Tirri H, Misikangas P, Sievanen J (2002) A probabilistic approach to wlan user location estimation. *Int J Wirel Inf Networks* 9(3):155–164
13. Cristianini N, Taylor J-S (2000) *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press, Cambridge
14. Matlab Website, <http://www.mathworks.com>