Available online at www.sciencedirect.com
ScienceDirect
ICT Express 2 (2016) 19–22
www.elsevier.com/locate/icte

Localization of networks with presence and distance constraints based on 1-hop and 2-hop mass–spring optimization☆

Gabriele Oliva[a,*], Roberto Setola[a], Stefano Panzieri[b], Federica Pascucci[b]

[a] University Campus Bio-Medico of Rome, Unit of Automatic, Via A. del Portillo 21, 00128 Rome, Italy
[b] Dipartimento di Ingegneria, University Roma TRE, Via della Vasca Navale, 79, 00146, Roma, Italy

Received 30 September 2015; received in revised form 14 January 2016; accepted 5 February 2016
Available online 23 February 2016

Abstract

In this paper we consider the localization of a sensor network where the nodes are heterogeneous, in that some of them are able to measure the distance from their neighbors, while some others are just able to detect their presence, and we provide a post-processing algorithm that can be used to improve an initial estimate for the location of the nodes, based on a mass–spring optimization approach, taking into account presence and distance information, as well as one-hop and two-hop information.

Keywords: Wireless sensor networks; Range-free localization; Trilateration; Mass–spring optimization

1. Introduction

The localization problem in sensor networks is usually addressed by considering nodes able to compute inter-distances (see for instance [1–3]), or sensors that are able to detect the presence of nodes in the neighborhood (e.g., [4]). In [5] we propose a different perspective, by considering hybrid networks, i.e., networks composed of both types of nodes. When the available information is affected by noise, however, the estimated position for the nodes might be unsatisfactory, and there is a need to provide adequate post-processing algorithms to reduce the position error. Among the others, the mass–spring optimization algorithm [6] shows good results in terms of error reduction and complexity.

In this paper, we extend the mass–spring optimization algorithm to hybrid networks, in order to handle both distance and presence information. Assuming the network is a unit disk graph (i.e., that a pair of nodes communicate when their distance is less than a given communication radius), moreover, we further improve the algorithm by taking into account also negative information on the fact that 2-hop neighbors (i.e., nodes that are not neighbors, but have a neighbor in common) are not connected. A simulation campaign which shows the benefits of the proposed approach concludes the paper.

The outline of the paper is as follows: in Section 2 we present the problem setting, while in Section 3 we review the mass–spring optimization algorithm; in Section 4 we develop an extension of the mass–spring optimization algorithm that takes into account also presence information and 2-hop information, while in Section 5 we present our simulations; some conclusive remarks are collected in Section 6.

2. Problem setting

Let us consider a *hybrid sensor network*, where some nodes, namely *presence nodes*, are able to measure just the presence of their neighbors, while some other nodes, namely *distance nodes*, are also able to measure the distance from their neighbors. We assume the distance nodes are able to transmit their measured distances to their presence neighbors, so a distance information is available for two sensors i and j provided that at least one of them is a distance node. The hybrid sensor network can be represented by a graph $G = \{V, E_d \cup E_p\}$ with $|V| = n$ nodes. The edges E_d and E_p represent *distance* and *presence*

* Corresponding author.
E-mail addresses: g.oliva@unicampus.it (G. Oliva), r.setola@unicampus.it (R. Setola), stefano.panzieri@uniroma3.it (S. Panzieri), federica.pascucci@uniroma3.it (F. Pascucci).

Peer review under responsibility of The Korean Institute of Communications Information Sciences.

☆ This paper is part of a special issue entitled “Positioning Techniques and Applications” guest edited by Prof. Sunwoo Kim, Prof. Dong-Soo Han, Prof. Chansu Yu, Dr. Francesco Potorti, Prof. Seung-Hyun Kong and Prof. Shiho Kim.

http://dx.doi.org/10.1016/j.icte.2016.02.005
2405-9595/

constraints, respectively. Let $p_i \in \mathbb{R}^2$ be the position of node $v_i \in V$.

A *distance constraint* is a constraint in the form $\|p_i - p_j\| = d_{ij} \leq \rho$, while a *presence constraint* is a constraint in the form $\|p_i - p_j\| \leq \rho$, where ρ is the communication radius and we assume that ρ is the same for all the agents. We assume that the graph G is a *unit disk graph*, i.e., a graph such that there is a link between two nodes v_i and v_j provided that $\|p_i - p_j\| \leq \rho$. The above assumption implies that we can use also *negative information* to get rid, to some extent, of position ambiguity. For instance, suppose that a localized node is not in reach of a non localized node; we conclude that the circle of radius ρ centered at the localized node does not contain the node to be localized. We assume the measured distances d_{ij} are affected by noise, and we assume we already have an estimate for the position of the nodes. In particular, we assume the nodes have calculated their position resorting to the approximated algorithms provided in [5]. We want to provide a mechanism to improve the accuracy of the localization of the nodes in the sensor network.

3. Mass–Spring optimization

Algorithm 1: Mass–Spring Optimization

```
t ← 0;
e(0) ← ∞;
p̂_i ← initial estimate for i = 1, ..., n;
p̂*_i ← p̂_i for i = 1, ..., n;
exit-condition ← 0;
while not exit-condition do
    calculate F_i(t), ∀i = 1, ..., n;
    p̂*_i ← p̂*_i + F_i(t)/(2|N_i|), ∀i = 1, ..., n;
    calculate e_i(t), ∀i = 1, ..., n;
    calculate e(t);
    if e(t) < e(t − 1) then
        p̂_i ← p̂*_i, ∀i = 1, ..., n;
    end if
    if e(t) < η then
        exit-condition= 1;
    end if
    t ← t + 1;
end while
return p̂_i for i = 1, ..., n;
```

In this section we briefly review the Mass–Spring Optimization technique [6], while we extend the framework in order to handle presence constraints and 2-hop information in the next section. The procedure described below is summarized in Algorithm 1. In [6] a mass–spring optimization algorithm is used to refine an initial estimate for the position of the nodes, assuming just distance constraints are available (i.e., $E_p = \emptyset$). Specifically, each link is treated as a spring whose natural length is the noisy measured distance $\hat{d}_{ij}(0) = d_{ij} + \delta_{ij}$ and the nodes v_i and v_j are initially estimated to be in the positions $\hat{p}_i(0)$ and $\hat{p}_j(0)$ that are the result of a localization procedure. The algorithm simulates a framework of springs and aims at reducing the energy associated to each node, in order to get close to a zero energy state, although in practice a local minimum is likely to be found [6]. Let $\vec{w}_{ij}(t)$ be the unit vector in the direction from $\hat{p}_i(t)$ to $\hat{p}_j(t)$, at time instant t. The force exerted by the single spring is given by

$$\vec{F}_{ij}(t) = \vec{w}_{ij}(t)(\hat{d}_{ij}(t) - \hat{d}_{ij}(0)) \tag{1}$$

where $\hat{d}_{ij}(t)$ is the distance calculated at step t as a result of the choice of $\hat{p}_i(t)$ and $\hat{p}_j(t)$.

The overall force for node i is given by

$$\vec{F}_i(t) = \sum_{j=1}^{n} \vec{F}_{ij}(t) \tag{2}$$

while the energy for node i is given by

$$e_i(t) = \sum_{j=1}^{n} e_{ij}(t) = \sum_{j=1}^{n} (\hat{d}_{ij}(t) - \hat{d}_{ij}(0))^2. \tag{3}$$

The total energy is calculated as

$$e(t) = \sum_{i=1}^{n} e_i(t). \tag{4}$$

At each step, the framework of springs is simulated in that each node moves its estimated position along the direction of $\vec{F}_i(t)$; such movement has a magnitude $|\vec{F}_i(t)|/(2|\mathcal{N}_i|)$, along the direction of the force $\vec{F}_i(t)$, where $\mathcal{N}_i$ is the number of 1-hop neighbors of node v_i, i.e., the number of nodes that are connected to v_i in the graph G. The above choice of the magnitude has been selected empirically in [6]. Notice that the movement is done if and only if the total energy is reduced.

The mass–spring algorithm amounts to a repetition of the above procedure, which is iterated until $e(t) < \eta$, for a given threshold η.

4. Mass–Spring optimization with presence and 2-hop information

4.1. Adding presence 1-hop information

Let us suppose an initial estimate $\hat{p}_i(0)$ for the position of each node v_i in an hybrid sensor network Σ is available. Differently from the standard Mass–Spring Optimization approach, we need to develop a mechanism to use the presence information.

For any two nodes v_i, v_j such that $(v_i, v_j) \in E$, we choose

$$\vec{F}^{one}_{ij}(t) = \vec{w}_{ij}(t) F^{one}_{ij}(t) \tag{5}$$

where

$$F^{one}_{ij}(t) = \begin{cases} 0, & \text{if } (v_i, v_j) \in E_p \text{ and } \hat{d}_{ij}(t) \leq \rho \\ \hat{d}_{ij}(t) - \rho, & \text{if } (v_i, v_j) \in E_p \text{ and } \hat{d}_{ij}(t) > \rho \\ \hat{d}_{ij}(t) - \hat{d}_{ij}(0), & \text{else.} \end{cases} \tag{6}$$

The above choice implies that the spring behaves as a regular spring, unless the link $(v_i, v_j) \in E_p$. In this case, in fact, the spring has a rest length equal to ρ; however we assume the