Gradient-Based Distance Estimation for Spatial Computers

Qingzhi Liu\textsuperscript{1,2,*}, Andrei Pruteanu\textsuperscript{1} and Stefan Dulman\textsuperscript{1}

\textsuperscript{1}Embedded Software Group, EEMCS Faculty, Delft University of Technology, Delft, The Netherlands
\textsuperscript{2}System Engineering Group, TPM Faculty, Delft University of Technology, Delft, The Netherlands
\textsuperscript{*}Corresponding author: q.liu-1@tudelft.nl

Today’s wireless networks are connecting more and more devices around us, leading to the birth of a new distributed computing platform, in the form of a spatial computer. The main difference with traditional computing models is that space and time become intertwined with computation, especially when scaling up the system. Computations performed by each element are now related to its spatial position. This property is the key ingredient when assuring the availability for various distributed networking services and applications. Computations become linked to the concept of space. Estimating distances between components (especially in dynamic networks characterized by the node mobility) thus becomes one of the most important building blocks for spatial computing. The majority of the algorithms that come from the MANET community presume knowledge about node position via systems such as GPS, or employ a one-time manual network topology configuration. While for some application scenarios this approach is feasible, for a lot of cases it suffers from frequent unavailability (e.g. indoors) and high costs in terms of energy consumption. Therefore, intense demand exists for a new kind of distance estimation algorithm using only simple local interactions, without knowledge of global information. The main contribution of the article is the introduction of a novel distributed algorithm, called gradient-based distance estimation (GDE), for the estimation of distances in networks characterized by mobility, specifically targeting the context of spatial computing. GDE is based on a gossiping mechanism to estimate distances between nodes with only local interactions, without knowledge of global information. The main contribution of the article is the introduction of a novel distributed algorithm, called gradient-based distance estimation (GDE), for the estimation of distances in networks characterized by mobility, specifically targeting the context of spatial computing. GDE is based on a gossiping mechanism to estimate distances between nodes with only local interactions. It significantly improves current state of the art by employing statistical analysis and making better use of the information available at each node. We analyze the parameters that should be considered by real applications, and present mathematical models to compensate their influence for distance estimation. Three spatial computing applications using GDE are presented: geographical cluster center detection, topological overlay shape construction and geographic routing. The simulation-based evaluation shows that GDE succeeds in estimating the distance between nodes in both static and mobile scenarios with considerably high accuracy for various simulations setups, such as varying node density, node speed or spatial node distribution.

Keywords: spatial computing; large-scale wireless networks; gradient; distance estimation; distributed algorithm

Received 10 April 2013; revised 26 September 2013
Handling editor: Vladimiro Sassone

1. INTRODUCTION

The spatial computing paradigm targets distributed systems that make use of spatial information in computational processes. Although this novel type of computer is intrinsically a distributed and parallel machine, its distributed and parallel characteristics are not only related to the distribution of data and processes, but also mainly to the spatial distribution of nodes and their interactions. There are already many kinds of distributed systems, such as wireless sensor networks, robotic swarms, engineered biological cells, etc., fitting these characteristics.

Distance estimation between components becomes naturally one of the most important building blocks for spatial computing. For the case of systems with static topologies (such as cellular automata alike) the connectivity between nodes can be determined at initialization time and updated during runtime to follow minor changes due to reconfigurations and failures (such
as the case of static wireless sensor networks). Determining distances in such cases is quite simple and a wide range of localization algorithms for static scenarios exist [1]. If the nodes become mobile, the problem requires more complex solutions. At the same time, quite a number of scenarios can be envisioned where GPS will be unavailable or provide a too coarse-grained accuracy. This is actually the operating scenario addressed by our work in this paper. This article presents a novel algorithm, called GDE, for estimating distances in large-scale spatial computers composed of mobile elements.

Nowadays, sensors are more and more widely used due to advances in low-power microelectronics and wireless network technology. They are used in many application domains, including environmental monitoring, smart buildings, target tracking, etc. As their number and diversity are increasing substantially, wireless sensor networks are being used to connect more and more devices around us. At the same time, more and more systems make use of services that are influenced by the spatial distribution of the nodes. In the recent years, the research community around the spatial computing paradigm has increased significantly. The current approaches to enable services such as routing, data dissemination or context detection, require new radical changes, in order to assure scalability and ability to cope with topology dynamics. When location information is required, most of the current algorithms presume that node position information is available via systems such as GPS. For some application scenarios, this approach is feasible (e.g. outdoors). Unfortunately, for most of the actual deployments, GPS is either not available (e.g. indoors) or it is inaccurate and consumes a lot of power. At the same time, in large-scale networks the sensor nodes cannot always be placed at desired locations, to ensure, for instance, a given network connectivity. Most likely, nodes also exhibit some sort of mobility, which makes continuously updating position information difficult. For all these reasons, there is a need for ways to estimate distances between nodes only based on local interactions with no additional information such as the one provided by GPS or any other classic localization techniques. On the other hand, availability of distance estimation enables many other applications for spatial computing.

In this article, we introduce a spatial gradient-based distance estimation algorithm called GDE, which is an extension from [2]. It is designed for large-scale spatial computers and uses simple local interactions. Compared with the existing gradient distance estimation algorithm [3], GDE significantly improves estimation accuracy by using new compensation methods based on statistical analysis of the hop count information available at each node. And it is further extended to compensate for the influence of various conditions that occur in real application scenarios, including varying node density, node speed, spatial node distribution, the percentage of successfully interactive neighbors, the duty cycling of the communication cycle, the shape of the transmission range, etc. We present a mathematical analysis as well as models for the influence of these parameters. The article is a theoretical study on the feasibility of such an approach for large-scale, dynamic systems. We validate our algorithms via simulations for various deployment setups, such as varying node density, different average node speed, realistic radio propagation models, etc.

The experimental results show that the proposed algorithm performs very well in terms of distance estimation accuracy. For static topologies with uniform distribution of nodes, GDE approaches the boundary of what is practically achievable, which takes estimation error three times smaller than the simple hop-count gradient (HCG) for high node densities. The simulation results show its strength in the more difficult scenarios, such as the non-uniform distribution of nodes, in which the error is reduced at most with a few orders of magnitude compared with the existing state of the art.

We employ GDE in three application scenarios: cluster center detection, overlay shape creation and geographic routing. Simulation results show that the applications using GDE exhibit an improved performance when compared with the other two distance estimation algorithms.

This article is organized as follows: Section 2 presents two alternative distance estimation algorithms. Section 3 introduces the basic GDE algorithm for a static, spatially uniform deployment. Section 4 presents the improved algorithm models for some realistic deployment parameters. An in-depth experimental analysis by means of simulations of the algorithms for various parameter setups is presented in Section 5. Section 6 shows how GDE can be used in three applications with improved performance. Before concluding the article and giving insights into future work in Section 8, we discuss the existing alternatives as related work in Section 7.

### 2. GRADIENT-BASED DISTANCE ESTIMATION

In this section, we introduce existing distance estimation algorithms based on gradients in networks [3, 4]. Our algorithm, GDE, builds upon these ideas and extends their performance and applicability.

#### 2.1. Hop-count gradient

Assume a network deployment where all nodes are static and uniformly distributed, and the transmission range of each node is \( r \). Unit disk graph communication is assumed for simplicity of presentation—extension to realistic communication scenarios is presented in Section 4.5. Node density is defined as the average number of nodes within the transmission range of each node. Some nodes, called seeds, are placed at fixed positions, and start with a constant counter value. Each node, except the seeds, listens to its neighbors and selects the largest counter value, and then rewrites the local counter value decreased by 1. The counter values of all nodes in the deployment area vary uniformly: the nodes closer to the seed nodes have larger counter values, and the
nodes that are further away from the seed nodes have smaller counter values. We refer to this variation trend as a gradient. Figure 1 shows the resulting gradient when a straight line of seed nodes is placed at the left-hand side of the deployment area. Figure 2 shows the resulting gradient when a seed node is placed at the center of the deployment area.

One may note in Figs 1 and 2 that the obtained gradient is actually discrete and not smoothed. This is caused by the fact that each node updates the local counter using the largest counter value in its own transmission range, and so, based on this strategy, some adjacent nodes will have the same local counter values. We call the nodes with the same counter values a wave in the gradient. Because the waves are determined by the minimum number of hops from the seeds, we call this gradient the HCG.

To create the HCG, the counter values should be disseminated in the network (gossiping algorithms [5] can be used for this).

2.2. Smoothed gradient

The distance estimation produced by the HCG is a multiple of the width of the gradient wave $\omega$. It produces an average error of $\sim0.5\omega$. To improve the accuracy, the local neighborhood gradient values can be used to compute the smoothed gradient (SMG) [3]. In the following, we exemplify this algorithm using a scenario where a dense line of seed nodes is placed at the border of the deployment area.

2.2.1. Regular nodes

Suppose that the transmission radius of each node is $r$. To simplify the computation of the SMG, the communication area
of every node is approximated to a square with side length 2r. Figure 3a shows the transmission area of node i. By a we denote the integer HCG value of node i, and α, β and γ are the ratios of three areas of various HCG waves as seen by node i. Owing to the fact that the nodes are uniformly distributed in the simulation area, α, β and γ are supposed to be equal to the percent of nodes in the adjacent three areas to the total number of nodes in the transmission range.

If the node density is infinite, then the maximum transmission distance of a node is actually r. The mechanism for creating HCGs leads to the width of each wave ω equaling r as well. In Section 3.1, we explains the reason of the width reduction in the gradient wave with low node density. The gradient value for each node is normalized to the length of the gradient wave.

The gradient distance of node i to the seed nodes is computed using Equations (1)–(3); Pa and Pb are the computed distances from node i to the line of seed nodes using parameters α and β, respectively; Ga and Gb are the gradient values; Paβ is the average value of Pa and Pb; and Gαβ is the gradient value.

\[
P_a = (a - 1)r + x \quad \text{and} \quad x = r - 2ra \Rightarrow G_a = P_a = a - 2a, \quad (1)
\]

\[
P_b = a - 1 + x \quad \text{and} \quad y = r - 2rb \Rightarrow G_b = P_b = a + 2b - 1, \quad (2)
\]

\[
P_{aβ} = \frac{P_a + P_b}{2} \Rightarrow G_{αβ} = G_{aβ} = a - α + β - \frac{1}{2}. \quad (3)
\]

The authors of [3] propose to average the gradient values of the neighboring nodes to compute the SMG (Equations (4) and (5)). The average gradient value is \(G_{avg}\) (in which \(n_i\) is the number of nodes in the transmission range of node i) and the SMG value is \(G_{SMG}\).

\[
G_{avg} = \frac{n_iα(a - 1) + n_i(1 - α - β)a + n_iβ(a + 1)}{n_i} = a - α + β, \quad (4)
\]

\[
G_{SMG} = G_{avg} - \frac{1}{2} = a - α + β - \frac{1}{2}. \quad (5)
\]

2.2.2. Border nodes

Equation (5) supposes that nodes in different HCG waves have different hop-count values. The nodes at the border of the HCG might discover nodes with non-initialized HCG values. Nodes in the neighborhood of the seed nodes can discover others belonging to different gradient waves while still having the same...
hop-count values. In Fig. 3b, the line of seed nodes takes the first HCG value of 0. Then, according to the HCG algorithm, nodes in area $\alpha$ and $\gamma$ have the same HCG value of 1. In Fig. 3c, gradient $a$ is the last HCG wave; then the nodes in area $\beta$ do not have a HCG value. Based on the equations given in [3], we present methods to correctly compute the SMG values of the nodes placed at the border of the HCG.

If the node is in the first HCG wave, as shown in Fig. 3b, the SMG value is given by Equation (6). If the node is located in the last HCG wave, as shown in Fig. 3c, the SMG value is given by Equation (7); $P_{\text{first}}$ and $P_{\text{last}}$ are the computed distances from node $i$ to the line of seed nodes in the first and last HCG wave, respectively; and $G_{\text{SMG}}^{\text{first}}$ and $G_{\text{SMG}}^{\text{last}}$ are the gradient values.

$$
\begin{align*}
P_{\text{first}} &= x \\
&= 2r\beta \\
P_{\text{last}} &= ar - x \\
&= 2r\alpha
\end{align*}
$$

![FIGURE 4. Width reduction of gradient waves.](http://comjnl.oxfordjournals.org/)

### 3. THE GDE ESTIMATOR

HCG and SMG gradient algorithms are designed based on a static and spatially uniformly distributed network. Paper [6] presents the characterization of the error in network gradients to quantify the distance estimation error, explores the causes of the error and models a part of discretization error. However, the work is based on an empirical model.

In this section, we present the GDE algorithm with analytic mathematical model to increase the accuracy of gradient distance estimation. GDE is based on the HCG and SMG algorithms that are introduced in Section 2. The final equation (Equation 25) results from three consecutive steps explored in the next three subsections: gradient width reduction, averaging of the reduced gradient width and the overestimated width reduction. Equation (25) incorporates all these steps and can be considered as the simple GDE estimator.

A common question when employing estimators focuses on the achievable accuracy of such approaches. In Section 3.4, we propose the derivation of an upper bound of the achievable accuracy.

#### 3.1. Step 1: Gradient width reduction

Suppose that node $i$ has a transmission range $r$ and node $k$ has the largest distance to node $i$ among all the nodes in the transmission range of node $i$. The number of nodes in the area being not infinite and the border being infinitely small lead to the fact that the distance between node $k$ and node $i$ is smaller than the transmission range $r$. In turn, this means that the width of the gradient wave in which $i$ and $k$ reside is smaller than $r$. This is referred to as the width reduction of gradient waves.

Figure 4 shows the phenomenon of width reduction of gradient waves with a line of seed nodes at the far left. Suppose that node $n$ is located at the border $p_{bu}$ of HCG wave $a - 1$, and node $m$ that is located at the border $p_b$ of HCG $a$ can just discover node $n$. Because node density is not infinite, there is a high chance that there is no node on the border $p_{bu}$, and the nodes that are closer to the border $p_{bu}$ are on the line $p_{wu}$. So the real border of the HCG wave $a - 1$ moves from $p_{bu}$ to $p_{wu}$. Suppose that node $n$ is moved to position $n'$. The last node that can discover node $n'$ has to move from position $m$ to $m'$. The border of the HCG wave $a$ changes from $p_b$ to $p_u$, because of the movement of the previous border from $p_{bu}$ to $p_{wu}$. The nodes positioned between $p_b$ and $p_u$, such as node $v$, can no longer discover the nodes with HCG value $a - 1$, so it updates the HCG value to $a + 1$.

At the same time, because of the nodes missing around border $p_u$, the nodes that are closer to the border $p_u$ are on the line $p$. The real border of the HCG wave $a$ moves from $p_u$ to $p$. As a result, the width of the HCG wave becomes smaller. At the same time, the movement of the border always influences the border placements of all the next HCG waves, so the border relocation, such as from $p_u$ to $p$ in HCG wave $a$, becomes larger and larger.

Suppose that the estimated distance corresponding to a communication hop is $d_{\text{hop}}$. Equation (8) (taken from [7]) shows the relation between the real estimated distance and the node density $n_i$.

$$
d_{\text{hop}} \leq \left( 1 + e^{-n_i} - \int_{-1}^{1} e^{-\left(n_i/\pi\right)(\arccos t - t\sqrt{1-t^2})} dt \right). \quad (8)
$$

Let $\Delta = 1 - d_{\text{hop}}$ be the width reduction of each HCG wave. As explained, the border relocation has a cumulative characteristic and is related to the HCG value. The cumulative reduced width of the wave with HCG value $a$ is $S_a = (a - 1)\Delta$ and the gradient width reduction value is

$$
G_{si} = G_{\text{SMG}} - S_i. \quad (9)
$$
3.2. Step 2: Averaging of reduced gradient width

The width reduction of the gradient wave introduced in Section 3.1 influences the result of the SMG. Each node in the HCG wave could be placed at two positions. The first place is shown in Fig. 5a, where node \( n \) is not in the reduced width area of the gradient wave. The second place is shown in Fig. 5b, where node \( n \) is in the reduced width area of the gradient wave.

In the first case, as shown by Fig. 5a, the percentages of discovered nodes in \( \alpha, \beta, \gamma \) become \( \alpha', \beta', \gamma' \), respectively, as shown in Equation (10).

\[
\begin{align*}
\alpha' &= \frac{2r\alpha - S_i^{a-1}}{2r} = 2\alpha - (a-1)\Delta/2 = \alpha - \frac{1}{2}(a-1)\Delta, \\
\beta' &= \frac{2r\beta + S_i^{a}}{2r} = \frac{2\beta + a\Delta}{2} = \beta + \frac{a}{2}\Delta, \\
\gamma' &= \frac{2r\gamma + S_i^{a-1} - S_i^{a}}{2r} = \frac{2\gamma + (a-1)\Delta - a\Delta}{2} = \gamma - \frac{1}{2}\Delta.
\end{align*}
\]

In the second case, as shown by Fig. 5b, the percentages of discovered nodes in \( \alpha, \beta, \gamma \) become \( \alpha'', \beta'', \gamma'' \), respectively, as shown in Equation (11).

\[
\begin{align*}
\alpha'' &= \frac{2r\alpha + (r - S_i^{a})}{2r} = \frac{2\alpha + (1-a)\Delta}{2} = \alpha + \frac{1}{2}(1-a)\Delta, \\
\beta'' &= \frac{2r\beta - (r - S_i^{a+1})}{2r} = \frac{2\beta - [1-(a+1)\Delta]}{2} = \beta - \frac{1}{2} + \frac{1}{2}(a+1)\Delta, \\
\gamma'' &= \frac{2r\gamma - (r - S_i^{a}) + (r - S_i^{a+1})}{2r} = \frac{2\gamma - (1-a)\Delta + [1-(a+1)\Delta]}{2} = \gamma - \frac{1}{2}\Delta.
\end{align*}
\]

If a node is outside the reduced width area, the average gradient value \( G_{\text{avg}}^{\text{out}} \) and the SMG value \( G_{\text{SMG}}^{\text{out}} \) are shown using Equation (12) and (13). If the node is inside the reduced width area, the average gradient value is \( G_{\text{avg}}^{\text{in}} \) and the SMG value becomes \( G_{\text{SMG}}^{\text{in}} \) as shown using Equation (14) and (15).

\[
\begin{align*}
G_{\text{avg}}^{\text{out}} &= \frac{n_1\alpha'(a-1) + n_1\gamma'\alpha + n_1\beta'(a+1)}{n_t} = a - 2\alpha + \left(\alpha\Delta - \frac{1}{2}\Delta\right), \\
G_{\text{SMG}}^{\text{out}} &= G_{\text{avg}}^{\text{out}} - \frac{1}{2} - (a-1)\Delta - \frac{1}{2}\Delta, \\
G_{\text{avg}}^{\text{in}} &= \frac{n_1\alpha''(a-1) + n_1\gamma''\alpha + n_1\beta''(a+1)}{n_t} = a - 2\alpha + \alpha\Delta - \frac{1}{2}\Delta, \\
G_{\text{SMG}}^{\text{in}} &= G_{\text{avg}}^{\text{in}} - \frac{1}{2} - (a-1)\Delta - \frac{1}{2}\Delta + 1.
\end{align*}
\]

Owing to the fact that nodes do not have position information and cannot tell whether they are inside or outside the reduced width area, we compute the expected average value of the gradients \( G_{\text{avg}}^{\text{out}} \) and \( G_{\text{avg}}^{\text{in}} \). Let \( R_{\text{in}} \) be the percentage of the reduced width of one HCG wave compared with the transmission range \( r \), as shown in Equation (16). Let \( R_{\text{out}} \) be the percentage of the not-reduced width of one HCG wave compared with the transmission range \( r \), as shown in Equation (17). The gradient value \( G_{\text{esti}} \) is

\[
R_{\text{in}} = \frac{(r - r_{\text{hop}})}{r} \equiv 1 - d_{\text{hop}},
\]

\[
R_{\text{out}} = \frac{(r - r_{\text{hop}})}{r} \equiv 1 - d_{\text{hop}}.
\]
The gradient computation presented in Section 3.1 uses the nodes on the border shown using Equation (19) [3]. For example, in Fig. 6b, node lines, the node density is overestimated.

3.3. Step 3: Overestimated width reduction

The gradient computation presented in Section 3.1 uses the expected average value \( d_{\text{hop}} \). This expected average value will overestimate the width reduction of gradient waves. Suppose that the gradient wave border moves as shown by Fig. 6a. If the border moves from \( p_b \) to \( p \), and node \( n \) is located in the area \( (p + d_{\text{hop}})^2 \), which means node \( n \) cannot discover the nodes on the border \( p \), then the computed hop distance \( d_{\text{hop}} \) is the expected width of the HCG wave. The real border that moves to \( p \) cannot be a straight line, but a curved line like \( p_r \). Node \( m' \) may move to the position of node \( m \), to be inside the transmission range of the node \( n \). As a result, node \( n \) can discover some of the nodes on the border \( p \). This implies that the reduced width \( \Delta \) is overestimated.

Assuming the shape of the gradient waves is a set of parallel lines, the node density is \( \rho \) and the transmission range of each node is \( r \). The probability that \( k \) nodes are located in area \( A \) is shown using Equation (19) [3]. For example, in Fig. 6b, node \( n \) has \( d_{\text{hop}} \) as the real estimated distance, which is introduced in Equation (8). Let \( p \) be the gradient border using \( d_{\text{hop}} \), \( p_d \) be the real border of the HCG, \( X \) be the overestimated reduced width and \( A \) be the overlap area of the overestimated reduced width area and the transmission range. The probability that there are no nodes in the area \( A \) equals the probability that the gradient border \( p \) will have a shift larger than \( x \), as shown using Equation (20). Therefore, the probability distribution function of the overestimated reduced width \( X \) is shown using Equation (21).

\[
R_{\text{out}} = 1 - R_{\text{in}} \frac{d_{\text{hop}}}{r} \quad (17)
\]

\[
G_{\text{esi}} = G_{\text{esi}}^{\text{out}} R_{\text{out}} + G_{\text{esi}}^{\text{in}} R_{\text{in}}. \quad (18)
\]

\[
\Pr(0 \in A) = \Pr(X > x) = e^{-\rho A}, \quad (20)
\]

\[
F(x) = \Pr(X \leq x) = 1 - e^{-\rho A}. \quad (21)
\]

The size of area \( A \) is computed using Equation (22) and the probability density function of \( X \) is shown by Equation (23). The expected overestimated width \( \Delta r \) of one HCG wave can be written as Equation (24). Let \( a \) be the HCG value of a node.

\[
A = 2 \left[ \pi r^2 \arccos((r - x)/r) - \frac{(r - x) \sqrt{r^2 - (r - x)^2}}{2} \right]
\]

\[
\equiv \arccos(1 - x) - (1 - x) \sqrt{2x - x^2}, \quad (22)
\]

\[
f(x) = F'(x) = (1 - e^{-\rho A}) dx
\]

\[
= \frac{\rho - 2x^2 + 4x - 1}{e(x-1) \sqrt{2x-x^2} + \rho \arccos(1-x) \sqrt{2x-x^2}}, \quad (23)
\]

\[
\Delta r = E(x) = \int_0^1 x f(x) dx
\]

\[
= \int_0^1 x \left( \frac{\rho - 2x^2 + 4x - 1}{e(x-1) \sqrt{2x-x^2} + \rho \arccos(1-x) \sqrt{2x-x^2}} \right) dx, \quad (24)
\]

\[
G_{\text{GDE}} = G_{\text{esi}}^{\text{out}} R_{\text{out}} + G_{\text{esi}}^{\text{in}} R_{\text{in}} + (a - 1) d_{\text{hop}} \Delta r. \quad (25)
\]

Equation (25) represents the basic GDE estimator. Its derivation was detailed progressively in the last three sections. For using it in practice, one needs to compute three values: the \( d_{\text{hop}} \) value based on Equation (8) (derived from the network density), \( G_{\text{SMG}} \) (given by Equation (5)) and \( \Delta r \) (derived in Equation (24)). These terms are the needed ingredients for obtaining a numerical value for the GDE estimator.

From an algorithmic perspective, the algorithm needs two pieces of information: an estimation of the density of the nodes in the network and information on the hop count of the neighboring nodes. It is worth noting that the GDE and SMG algorithms use exactly the same pieces of information.
3.4. Upper boundary on achievable precision

In this section, we introduce an error reference value, to be used as a benchmark for the testing results of the distance estimation errors.

If node density \( \rho \) tends to \( +\infty \), then \( \Delta \) and \( \Delta r \) (introduced in Sections 3.1 and 3.3) converge to 0, as shown by Equations (26) and (27). This means that the gradient width reduction and the gradient width overestimation problems disappear for large node densities.

\[
\lim_{\rho \to +\infty} \Delta r = 1 - \lim_{\rho \to +\infty} d_{\text{hop}} = 0, \quad (26)
\]

\[
\lim_{\rho \to +\infty} F(x) = \lim_{\rho \to +\infty} (1 - e^{-\rho A}) = 1,
\]

\[
\Rightarrow \lim_{\rho \to +\infty} f(x) = 0 \Rightarrow \lim_{\rho \to +\infty} \Delta r = 0. \quad (27)
\]

The number of nodes \( n \) in the area of size \( h \) follows a Poisson distribution with parameter \( \lambda = \rho \cdot h \). If node density \( \rho \) tends to \( +\infty \), then the probability of \( n \) nodes in the area of size \( h \) tends to follow a Normal distribution with a mean \( \lambda \) and a deviation of \( \sqrt{\lambda} \), as shown using Equation (28). Therefore, if node density \( \rho \) goes to \( +\infty \), the number of nodes in various HCG waves of one transmission range also obeys a Normal distribution, as shown in Equation (29), in which \( n_{\alpha} \) and \( n_{\beta} \) are the number of nodes in sections \( \alpha \) and \( \beta \), and \( S_{\alpha} \) and \( S_{\beta} \) are the area size of \( \alpha \) and \( \beta \), respectively.

\[
n \sim P(\lambda) = \frac{\lambda^n}{n!} e^{-\lambda} \Rightarrow n \sim N(\lambda, \lambda), \quad (28)
\]

\[
n_{\alpha} \sim N(\rho S_{\alpha}, \rho S_{\alpha}) \quad n_{\beta} \sim N(\rho S_{\beta}, \rho S_{\beta}). \quad (29)
\]

The linear combination of Normal distributions is also a Normal distribution. The computed gradient value using Equation (25) follows a Normal distribution as shown using Equation (30), in which \( n_{l} \) is the average number of nodes in the transmission range of a node.

\[
\lim_{\rho \to +\infty} G_{\text{GDE}} = \lim_{\rho \to +\infty} \left( G_{\text{out}}^{\text{out}} R_{\text{out}} + G_{\text{in}}^{\text{in}} R_{\text{in}} \right)
\]

\[
= \lim_{\rho \to +\infty} \left( a - \alpha + \beta - \frac{1}{2} \right)
\]

\[
= \lim_{\rho \to +\infty} \left( a - \frac{n_{\alpha}}{n_{l}} + \frac{n_{\beta}}{n_{l}} - \frac{1}{2} \right)
\]

\[
\sim N \left( a - \frac{\rho S_{\alpha}}{n_{l}} + \frac{\rho S_{\beta}}{n_{l}} - \frac{1}{2}, \frac{\rho (S_{\alpha} + S_{\beta})}{n_{l}^2} \right). \quad (30)
\]

If nodes are distributed on a grid and the node density goes to infinity, then, by using Equation (31), the gradient is computed as \( G_{\text{GDE}}^{\text{GDE}} \). If nodes in the area are uniformly distributed and the node density goes to infinity, then using Equation (32), the gradient is computed as \( G_{\text{GDE}}^{u} \).

\[
G_{\text{GDE}}^{\text{GDE}} = a - \frac{\rho S_{\alpha}}{n_{l}} + \frac{\rho S_{\beta}}{n_{l}} - \frac{1}{2}, \quad (31)
\]

\[
G_{\text{GDE}}^{u} = a - \frac{n_{\alpha}}{n_{l}} + \frac{n_{\beta}}{n_{l}} - \frac{1}{2}. \quad (32)
\]

If all nodes are uniformly distributed, the error becomes \( \Delta G \), which is the absolute difference value between the gradient that is computed when nodes are distributed on a grid and the gradient computed when nodes are uniformly distributed at random locations, as shown using Equation (33), in which \( S_{l} \) is the size of the transmission area.

\[
\Delta G = G_{\text{GDE}}^{\text{GDE}} - G_{\text{GDE}}^{u} = a - \frac{n_{\alpha}}{n_{l}} + \frac{n_{\beta}}{n_{l}} - \frac{1}{2} - \left( a - \frac{S_{\alpha}}{S_{l}} + \frac{S_{\beta}}{S_{l}} - \frac{1}{2} \right)
\]

\[
= \left( \frac{S_{\alpha}}{S_{l}} - \frac{n_{\alpha}}{n_{l}} \right) + \left( \frac{S_{\beta}}{S_{l}} - \frac{n_{\beta}}{n_{l}} \right). \quad (33)
\]

Equation (29) shows the probability distribution of the number of nodes \( n_{\alpha} \) and \( n_{\beta} \), while \( \Delta G \) is the linear combination of the Normal distributions of \( n_{\alpha} \) and \( n_{\beta} \). The error \( \Delta G \) also follows a Normal distribution, as shown using Equation (34).

\[
\Delta G \sim N \left( \frac{S_{\alpha}}{S_{l}} - \frac{\rho S_{\alpha}}{n_{l}^2}, \frac{\rho S_{\alpha}}{n_{l}^2} \right) + N \left( \frac{S_{\beta}}{S_{l}} - \frac{\rho S_{\beta}}{n_{l}^2}, \frac{\rho S_{\beta}}{n_{l}^2} \right)
\]

\[
= N \left( \frac{0, \rho S_{\alpha}}{n_{l}^2} \right) + N \left( 0, \frac{\rho S_{\beta}}{n_{l}^2} \right)
\]

\[
= N \left( 0, \frac{\rho (S_{\alpha} + S_{\beta})}{n_{l}^2} \right) \sim N \left( 0, \frac{1}{2n_{l}} \right). \quad (34)
\]

The average absolute error for all nodes is the expected value for \( |\Delta G| \). \( Z \) is the standard Normal distribution of \( \Delta G \) within function domain \([0, +\infty)\); and \( a \) and \( \delta \) are the mean value and the standard deviation of the Normal distribution \( \Delta G \), respectively. The expected value of \( Z \) is presented by Equation (35).

\[
E(Z) = \frac{1}{\sqrt{2\pi}} \int_{0}^{+\infty} z e^{-z^2/2} dz = \frac{1}{\sqrt{2\pi}}. \quad (35)
\]

We see that \( \Delta G \) is the linear function of \( E(Z) \), therefore the expected absolute error of distance estimation is

\[
E(|\Delta G|) = 2E(\delta Z) = 2\delta E(Z)
\]

\[
= \frac{2}{\sqrt{2n_{l}}} \frac{1}{\sqrt{2\pi}} = \frac{1}{\sqrt{\pi n_{l}}}. \quad (36)
\]

If the node density goes to infinity, the distance estimation error goes to zero. In reality, node density cannot be infinite. Thus, the previous result is the error of distance estimation in the ideal case, which can be used as a benchmark for the experimental results.
4. RELAXATION OF GDE ASSUMPTIONS

So far we have discussed algorithms in more or less idealized conditions. To apply the gradient algorithm in the real-world cases, Beal [8] presents the algorithm to take advantage of the tradeoff between estimation precision and communication cost to decrease the energy cost in the communication of creating gradient. For real deployments, wireless systems have to cope with many other problems such as node mobility, communication failures, non-uniform node distributions, influence of different communication paradigms (unicast vs. multicast), etc. In this section, we make a mathematical analysis model for these factors and present methods to make the GDE algorithm adapt to various parameters.

4.1. Approximation error

The equations used by the GDE algorithm to compute the gradient presume that all nodes are uniformly distributed in the area and the transmission area of each node is a square. The equations that compute the overestimated width reduction of the gradient in Section 3.3 consider a given node distribution, but the computations assume that the shape of the HCG is a set of parallel lines. In reality, the transmission range of the nodes is more like a circle. Most of the time only one or a small number of nodes can be set as the seed nodes. The shape of the HCG waves can also be a circle with the seed node at the center. In the previous computations, the transmission range is approximated to a square and the parallel gradient waves are considered to be parallel lines. The approximation of nonlinear conditions with linear conditions introduces errors. This section explains the error introduced by the use of a parallel gradient shape instead of circular gradient shape. At the same time, we analyze some characteristics of the nonlinear conditions, and present an approach to repair these errors.

4.1.1. Circular gradient

If the HCG shape is circular instead of parallel lines, the overestimated gradient shrink compensation value $\Delta r$ that is computed as shown in Section 3.3 still holds but the size of the overlapping area is different. In Fig. 7, $o$ is the place of the seed node, $n$ is a node with real transmission radius $d_{hop}$, $K$ is one circle border of the HCG wave without considering the overestimated wave width reduction, and $K'$ is the border considering the overestimated wave width reduction. If the circular gradient border $K$ and the compensated border $K'$ have overlapping areas with the transmission range of node $n$, then they are represented by areas $a_1$ and $a_2$. If there is a line of seed nodes vertically placed at position $o$, then the gradient border $K$ will become $p$ and the compensated border $K'$ will become $p_d$. The overlapping area of the gradient borders $p$, $p_d$ and the transmission range of node $n$ is almost equal to the overlapping areas $a_1$ and $a_2$ which are created by the circle borders $K$, $K'$ and the transmission range. With the increasing of the radius of the gradient wave circle, the circular gradient wave lines $K$ and $K'$ will be closer and closer to the parallel wave lines $p$ and $p_d$. The difference of the areas between the circular waves and the parallel waves will become smaller and smaller. If the radius of the circular gradient wave is infinite, then the gradient border can be seen as parallel lines. If the radius of the circular gradient wave is small, then the overestimated shrink compensation value $\Delta r$ will be overestimated.

We present the detailed computation method for the overestimated width reduction of the circular gradient waves. As shown by Fig. 7, $a_1$ and $a_2$ are the two parts of the overlapped areas, $x_1$ and $x_2$ are the width of the two overlapped areas, and $p'$ is a line used to split the overlapping area into $a_1$ and $a_2$. Area $A$, which is computed by Equation (22), is composed of the area $a_1$ and $a_2$. Suppose that the length $x$ is the sum of length $x_1$ and $x_2$. $R$ is the radius of the circular gradient wave $K$ and $r$ is the transmission range which equals $d_{hop}$. The relation between $x_1$ and $x$, $a_1$ and $x_1$, $a_2$ and $x_2$ are given by Equations 37–(39).

When the transmission range $r$ is normalized to 1, the radius $R$ can be replaced by the gradient value $G_i$ of node $i$. Since the gradient difference between node $n$ and node $i$ is 1, we use $G_i = G_n + 1$ to compute the gradient value of node $i$, in which $G_n$ is the gradient value of node $n$.

\[
\begin{align*}
(R + x)^2 - (R + x_1)^2 &= r^2 - (r - x_1)^2 \\
x_1 + x_2 &= x \\
\Rightarrow x_1 &= \frac{R}{R + r} x_{r=1,R=G_i} \\
\Rightarrow x_1 &= \frac{G_i}{G_i + 1} x, \quad (37) \\
a_1 &= \arccos(1 - x_1) - (1 - x_1)\sqrt{2x_1 - x_1^2}, \quad (38)
\end{align*}
\]
The relation between the radius of the gradient circle and the length of the crossed arc of the gradient circle and the transmission range.

\[ a_2 = 2 \left[ \frac{\pi (R + x)^2 \arccos((R + x_1)/(R + x))}{2\pi} - \frac{(R + x_1)\sqrt{(R + x)^2 - (R + x_1)^2}}{2} \right] \]

\[ r = 1, R = G_i \]

\[ \Rightarrow (G_i + x_1)^2 \frac{G_i}{G_i + x} \]

\[ \frac{G_i}{G_i + x} \]

\[ (G_i + x)^2 \frac{G_i}{G_i + x} \frac{G_i + (G_i/(G_i + 1))x}{G_i + x} \]

\[ A(x) = a_1 + a_2 = \arccos \left( 1 - \frac{G_i}{G_i + 1} \frac{x}{x} \right) \]

\[ d_1 = d + d_2 = d \]

\[ \Delta r = E(x) = \int_0^1 x f(x) dx = \int_0^1 x (1 - e^{-\rho A(x)}) dx. \quad (41) \]

4.1.2. Circular transmission range

The equations provided in the previous section suppose the transmission range to be square in order to simplify the computation. Equation (41) recomputes the overestimation of the width reduction. The error of the linear approximation of the square transmission range in the other equations can also introduce an error for distance estimation. Here, we present a method to take advantage of the nonlinear characteristic of the gradient wave and circle transmission range to repair the error of the distance estimation.

Suppose that the gradient waves have circular shape and the transmission range of each node is also a circle as shown in the Fig. 8a, in which \( g_1 \) and \( g_2 \) are two gradient lines that have the same distance to node \( n \). The length of the crossed arcs of two gradient lines \( g_1 \) and \( g_2 \) and the transmission range of node \( n \) are different.

Next, we present the computation of the length of the crossed arc of the gradient line and the transmission range of node \( n \). As shown in Fig. 8b, \( o \) is the seed node and \( n \) is a node placed in the gradient, the circle with radius \( R \) is one of the gradient circles, the circle with radius \( r \) is the transmission range of node \( n \), \( d_1 \) is the distance from \( o \) to line \( c \) and \( d_2 \) is the distance from \( n \) to line \( c \). Suppose that \( d \) is the distance between the seed node \( o \) and node \( n \), \( l \) is the length of the crossed arc of the gradient circle and the transmission range. The length of the crossed arc \( l \) of the gradient line and the transmission range of node \( n \) is given by

\[ r^2 - d_1^2 = R^2 - d_1^2 \]

\[ d_1 + d_2 = d \]

\[ = 2R \arccos \frac{R^2 - r^2 + d^2}{2Rd} (d - r \leq R \leq d + r). \quad (42) \]
Equation (42) shows that if the transmission range $r$ and the distance $d$ are fixed values, then the length of arc $l$ is a function of radius $R$. Figure 8c shows the relation between the radius of the gradient circle and the length of the crossed arc. In Fig. 8c, we fix the value of $r$ to 1 and $d$ to 2, and the $x$ coordinate is the radius of the gradient circle and $y$ coordinate is the length of the crossed arc of the gradient circle and the transmission range. If two gradient circles have the same distance to node $n$, then the gradient circle with larger radius $R$ has a larger arc value $l$. It means that, in the same parts of the deployment area, the probability that the nodes on the gradient circle with smaller radius is more likely to be not discovered than on the gradient circle with larger radius.

Suppose that there are two nodes $n_{\text{close}}$ and $n_{\text{far}}$, and both nodes $n_{\text{close}}$ and $n_{\text{far}}$ are in the transmission range of node $n$. Node $n_{\text{close}}$ is the node that is closest to the seed node among the nodes in the transmission range of node $n$, and node $n_{\text{far}}$ is the node that is the furthest from the seed node among the nodes in the transmission range of node $n$. Then $n_{\text{close}}$ is on the gradient circle with smaller radius, while $n_{\text{far}}$ is on the gradient circle with larger radius. So, according to the Equation (42), the probability that node $n_{\text{close}}$ is not discovered by node $n$ is larger than the probability that node $n_{\text{far}}$ is not discovered by node $n$. This conclusion also means that, from the probabilistic point of view, the distance from node $n$ to node $n_{\text{far}}$ is larger than the distance from the node $n$ to the node $n_{\text{close}}$, since the theoretical closest node $n_{\text{close}}$ is more likely to be discovered than the one placed at a high distance $n_{\text{far}}$.

Algorithm 1 presents the detailed steps required to repair the error of the circle transmission range in the circle gradients. Suppose that the gradient values become larger and larger from the seed nodes; $G_{\text{GDE}}$ is the GDE gradient value of node $n$; $G_{\text{GDE}}^{\min}$ is the minimum GDE gradient value of the neighbors, which means the node is the closest to the seed node among all neighbors of node $n$; $G_{\text{GDE}}^{\max}$ is the maximum GDE gradient value of the neighbors, which means the node is the most distant from the seed node among all neighbors of node $n$. Suppose that the gradient difference between $G_{\text{GDE}}$ and $G_{\text{GDE}}^{\max}$ is statistically larger than the difference between $G_{\text{GDE}}$ and $G_{\text{GDE}}^{\min}$. If the gradient difference between $G_{\text{GDE}}$ and $G_{\text{GDE}}^{\min}$ is statistically larger than the difference between $G_{\text{GDE}}$ and $G_{\text{GDE}}^{\max}$, then we recomputed the gradient value of node $n$ to $G_{\text{GDE}}^{\text{repair}}$.

The algorithm provides a method to repair the distance estimation value from the statistical point of view. It is true that, because of the random distribution of nodes in the area, there exists the possibility that the node distance estimation is correct but $|G_{\text{GDE}} - G_{\text{GDE}}^{\min}| > |G_{\text{GDE}}^{\max} - G_{\text{GDE}}|$. The Algorithm 1 will introduce error to the gradient value. Statistically, the error introduced is always smaller than the error decreased by Algorithm 1. At the same time, if the distance $d$ is much larger than the radius $r$, the gradient shape is almost like parallel lines, the difference between the arc lengths that are closer to and further from the seed node does not exist. So

<table>
<thead>
<tr>
<th>Algorithm 1 : Repair GDE gradient value statistically.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong></td>
</tr>
<tr>
<td>GDE gradient value</td>
</tr>
<tr>
<td><strong>Output:</strong></td>
</tr>
<tr>
<td>Repaired GDE gradient value</td>
</tr>
<tr>
<td><strong>Parameters:</strong></td>
</tr>
<tr>
<td>$G_{\text{GDE}}$</td>
</tr>
<tr>
<td>$G_{\text{GDE}}^{\min}$</td>
</tr>
<tr>
<td>$G_{\text{GDE}}^{\max}$</td>
</tr>
<tr>
<td>$\triangleright$ Minimum gradient value in neighbors</td>
</tr>
<tr>
<td>$\triangleright$ Maximum gradient value in neighbors</td>
</tr>
</tbody>
</table>

If $|G_{\text{GDE}} - G_{\text{GDE}}^{\min}| > |G_{\text{GDE}}^{\max} - G_{\text{GDE}}|$ then
$$
G_{\text{GDE}}^{\text{repair}} = (G_{\text{GDE}}^{\min} + G_{\text{GDE}}^{\max})/2
$$
end if

Algorithm 1 only works for gradients built for small network diameters.

### 4.2. Node mobility

In Section 3, we introduce the GDE algorithm in the static environment. However, in most cases of the application, nodes are dynamic. In this section, we provide a method to compensate the influence of mobile nodes to the distance estimation of the GDE algorithm.

In Fig. 9a, $p_b$ is the border of a HCG wave. Node $m$ can travel from position $p$ to $p'$ and $p''$ in one time step. The last node that can discover the border $p_b$ changes from node $n$ to node $n'$, which enlarges the width of the HCG wave of the gradient. A strategy to compensate the enlarged width in the distance estimation value is needed.

We use the expected moving distance of one node in a given time step $E_{dc}$ to compensate the enlarged width value, which equals the movement distance from $p_b$ to $p'$. The translation of the HCG into $x$ and $y$ coordinates is shown by Fig. 9b, in which the $y$ coordinate is parallel to the line of the seed nodes.

When nodes move with various speeds, the average expected moving distance for a node in one time step is $D$, which makes an angle $\theta$ with the $y$-axis ($V$ being the projection on it). Owing to the fact that all the possible movement directions are symmetrical with respect to the $x$ coordinate, only the movement for $y > 0$ is considered ($\theta$ ranging from $-\pi/2$ to $\pi/2$). The probability density function of $V$ is given by Equation (43). Since we need the absolute expected moving distance of one node, the expected value is integrated from 0 to $D$ as shown by Equation (44).

$$
\theta \sim U\left(-\frac{\pi}{2}, \frac{\pi}{2}\right) \Rightarrow f(v) = \frac{1}{\pi\sqrt{D^2 - v^2}}, \quad (43)
$$

$$
E_{dc} = E(v) = \int_0^D v f(v) dv = \int_0^D \frac{v}{\pi\sqrt{D^2 - v^2}} dv. \quad (44)
$$
The enlarged width of each gradient wave is accumulating from one HCG to the next. For a node with HCG value \( a \) and transmission range \( r \), the dynamic compensation value \( C_d \) and the gradient value \( G_{\text{GDE}}^{\text{comp mob}} \) are

\[
C_d = \frac{[(a - 1)d_{\text{hop}}E_{dc}]/r, \quad (45)}{\sqrt{1-x^2}}, \quad -1 < x < 1, \quad \text{otherwise},
\]

\[
G_{\text{GDE}}^{\text{comp mob}} = G_{\text{GDE}} + C_d. \quad (46)
\]

The computations above suppose that all nodes have the same constant speed \( D \) for different moving directions. In practical applications, all nodes move with different speeds. Here, we present two methods to compensate the distortion produced by node mobility.

The first method supposes that we know the average expected speed of the nodes \( E(D) \). We replace the average expected moving distance for a node in one time step \( D \) with \( E(D) \) in the Equation (43), while keeping all the other computations the same as in Equations (44)–(46).

The second method supposes that we know more detailed conditions about the node speeds, such as the probability density function of all nodes. The computation of the expected movement distance can be used to compensate the movement of nodes. Suppose that \( \theta \) is the moving direction of the node, \( X \) is the function of \( \theta \) and \( \theta \) is uniformly distributed from \(-\pi/2\) to \( \pi/2\); then the probability density function of \( X \) is shown in Equation (47). Suppose that \( Y \) is the speed of the nodes following an uniform distribution from 0 to \( D \). The probability density function of \( Y \) is computed in Equation (48).

The relation between \( V \), \( X \) and \( Y \) is shown in Equation (49). Suppose that \( V \) is the moving distance to the positive direction of \( x \). The relation between \( V \), \( X \) and \( Y \) is shown in Equation (49), in which \( X \) and \( Y \) are mutually independent from each other [9]; the probability density function of \( V \) is given by Equation (50) and the average expectation value of \( v \) by Equation (51).

\[
\theta \sim U\left(-\frac{\pi}{2}, \frac{\pi}{2}\right) \Rightarrow f(x) = \begin{cases} \frac{1}{\pi \sqrt{1-x^2}}, & -1 < x < 1, \\ 0, & \text{otherwise}, \end{cases} \quad (47)
\]

\[
Y \sim U(0, D) \Rightarrow f(y) = \begin{cases} \frac{1}{D}, & 0 < y < D, \\ 0, & \text{others}, \end{cases} \quad (48)
\]

\[
V = Y \sin \theta = XY, \quad (49)
\]

\[
f(v) = \int_{-\infty}^{+\infty} \frac{1}{|x|} f\left(x, \frac{v}{x}\right) dx = \int_{-\infty}^{+\infty} \frac{1}{|x|} f_X(x) f_Y \left( \frac{v}{x} \right) dx = \int_{v/D}^{1} \frac{1}{|x|} \frac{1}{\pi \sqrt{1-x^2}} \frac{1}{D} dx = -\frac{1}{\pi D} \ln \frac{D - \sqrt{D^2 - v^2}}{v}, \quad 0 < v < D, \quad (50)
\]

\[
E_{dc} = E(v) = \int_{0}^{D} v f(v) dv = -\frac{1}{\pi D} \int_{0}^{D} v \ln \frac{D - \sqrt{D^2 - v^2}}{v} dv. \quad (51)
\]
parameter $Y$ also follows the uniform distribution. In real applications, the mobility pattern of nodes and distribution functions of some parameters are different. But the enlarged width of the HCG wave caused by the mobility of nodes will still exist. Even in this case, the compensation strategy and the computation process provided above still decrease the error caused by the node mobility.

4.3. Multicast communication

The GDE algorithm does not require nodes to communicate with all their neighbors (defined here as broadcast communication). More often, if nodes only choose a fraction of the neighborhood to communicate with, we define this as multicast communication. If communication is further reduced to selecting only one neighbor, we refer to it as unicast communication.

Suppose that the nodes are uniformly distributed in the area and each node only selects a fraction of its neighbors to communicate with. Compared with the original network, the new network has an average lower node density. According to the equations in Section 3, GDE can estimate distances for networks with different node densities. Therefore, each node can use multicast to communicate with their neighbors. The equations of GDE algorithm introduced in Section 3 still hold, while the only difference is that node density $n_l$ needs to be adjusted with the percentage of neighborhood size considered.

At the same time, GDE uses gossiping to reduce the message complexity compared with flooding. In a fixed network with $n$ nodes, if a single message needs to be spread using randomized gossiping, the number of rounds required for converging is $O(\log n)$ and the message complexity converges to $O(n \log n)$ [10]. If each node in the network broadcasts the message to the other nodes in the network, the worst case of the message complexity becomes $O(n^2)$, which is much larger than the gossip-based information dissemination.

4.4. Round length

GDE assumes a discrete time model, where communication rounds of size $P$ are not synchronized. Each node acts once in each round, in a different time slot. The rounds are not synchronized and are being used to mimic the asynchronous nature of communication. Suppose that all nodes follow a Random Walk Mobility model, the transmission range is $r$ and the expected moving distance at one time step is $E_{dc}$. We model the motion segments with a Markov Chain in order to compute the expected moving distance (see Fig. 10). We only consider nodes moving on one dimension (i.e. toward the next HCG). Suppose that states $j, \ j = 0, \ldots, N,$ describe a node moving forward for a distance of $j \cdot E_{dc}$. By $p_j$, we represent the probability that the node moves further from the original position, while $q_j$ is the probability of a node moving closer to the original position. Let $N$ be the maximum number of moving steps in one round of length $P$. Let $r_0$ be the probability that a node does not move out of the initial state and is equal to $\frac{1}{2}$, and let $p_0$ be the probability that a node moves out of the initial state and is equal to $\frac{1}{2}$. In the inner processes, the backward and forward motion probabilities of a node in one time step are the same; so the node moves to the next state or to the previous state with probability $p_j = q_j = \frac{1}{2}$. For the final state $N$, due to the fact that there are nodes that can move to the previous state or start a new computation round, let $r_N$ be the probability that a node starts a new round (equal to $\frac{1}{2}$), and $q_N$ be the probability that a node moves backward (equal to $\frac{1}{2}$). Owing to the fact that there is a large number of nodes, the probability of each steady state is the probability that a node stays in that state for $P$ steps. Considering the mathematical properties of the Random Walk [11], the model that describes the steady states $\pi_j$ ($0 \leq j \leq N$) is given by Equation (52). One can note that all states converge to the same probability.

$$
\pi_0 = r_0 \pi_0 + q_1 \pi_1 \Rightarrow \pi_0 = \pi_1,
$$

$$
\pi_N = p_{N-1} \pi_{N-1} + r_N \pi_N \Rightarrow \pi_N = \pi_{N-1},
$$

$$
\pi_j = \frac{\prod_{m=0}^{j-1} (p_m/(q_{m+1}))}{1 + \sum_{j=0}^{N} \prod_{m=0}^{j-1} (p_m/(q_{m+1}))} \Rightarrow p_{\text{avg}} = \frac{1}{1 + N} (1 \leq j < N). \tag{52}
$$

The node in state 0 does not affect the gradient of nodes in the next HCG. For this reason, we ignore state 0. For each state in the Markov Chain, it represents the expected moving distance from the original place. For state $j$, the moving distance is $j \cdot E_{dc}$. Because every steady state has the same probability as shown by Equation (52), the expected value for the moving distance of all states is given by Equation (53). If the length of the round becomes 1, Equation (53) is the same as Equation (44). The compensation value for the round length is given by Equation (54), and the computation for the gradient value is given by Equation (55).

$$
E_{dc} = \frac{\sum_{j=1}^{P} j E_{dc}}{P} = \frac{(1 + P) E_{dc}}{2}, \tag{53}
$$

FIGURE 10. Node moving states of each time step of a round in the Markov Chain.
We use the average transmission range to create the HCG, each node discovers the largest gradient that follows a normal distribution. If the node is between the average transmission radius of each node and the transmission range follows a normal distribution, the gradient value is not a constant value and has rather different shapes according to the application conditions. In this section, we present a compensation strategy to allow the GDE algorithm to work with irregular transmission ranges.

4.5. Realistic model for the transmission range

In the previous sections, the transmission range uses the unit disk graph model. In real applications, the transmission range is not a disk, and has rather different shapes according to the application conditions. In this section, we present a compensation strategy to allow the GDE algorithm to work with irregular transmission ranges.

First, we presume that the communication range of a node is not a constant value \( r \), but follows a Normal distribution. To create the HCG, each node discovers the largest gradient value in its transmission range. So the HCG value for each node depends on the largest transmission range the node can have. If the transmission range follows a Normal distribution \( N(\lambda, \delta^2) \), the mean value \( \lambda \) is not the highest distance a node can transmit. We use the average transmission range \( \lambda + \delta \) as the new radius.

In the equations introduced in the previous sections, we use the mean value \( \lambda \) as the width of the wave of the HCG to compute the gradient value. For every wave of the HCG, the distance is underestimated with a value of \( \delta \).

As shown in Fig. 11, the transmission range of node \( n \) follows Normal distribution \( N(\lambda, \delta^2) \). Suppose that node \( n \) is at the border of the HCG wave \( a \). The next hop-count wave border changes from \( b \) to \( b' \), due to the Normal distribution of transmission distance. If the node is in between gradient border \( a \) and \( b \), then the underestimated distance caused by the previous HCG waves \( D_p \) is given by Equation (56), in which \( \lambda \) is the average transmission radius of each node and \( \delta \) is the standard deviation of the transmission radius. If the node is between the gradient border \( b \) and \( b' \), then the underestimated distance caused by its own HCG wave is given by Equation (57). The computation of the gradient value is given in Equation (58).

\[
D_p = (G_{HCG} - 1)\lambda, \quad (56)
\]
\[
D_l = \frac{\delta^2}{2(\lambda + \delta)} = \frac{\delta^2}{2(\lambda + \delta)}, \quad (57)
\]
\[
G_{\text{GDE round}}^{\text{comp trans}} = G_{\text{GDE}} + [D_p + D_l]/r. \quad (58)
\]

This section only uses Random Walk as a mobility model. For the other mobility models, such as SLAW [12], the probabilities of the states on the Markov Chain take different values, although the analysis and computation process remain similar.

4.6. Integrated parameters

For real deployments, the spatial node distribution is not uniform. Furthermore, node speed, the neighborhood interaction and the communication rounds vary. Therefore, we introduce an algorithm that is able to compute the gradient value for nodes for various deployment conditions as shown by Algorithm 2.

Each node communicates with a fraction of its neighbors to compute its own gradient values as shown in Section 3, and propagates gradient information to the nodes belonging to the next HCG wave. The propagated gradient information includes average values of the gradient reduced width value \( \Delta \), the static compensation value \( E_{sc} \) and the dynamic compensation value \( E_{dc} \). Each node uses the received width reduction values and compensation values to compute its own gradient value.

For Algorithm 2, \( G_{HCG} \) is the HCG value, \( \{SN\} \) is the subset of the neighbors to communicate with, \( d_{hop} \) is the real one hop distance, \( E_{sc} \) is the compensation for the static case which equals \( \Delta r \) as shown in Equation (24) and \( E_{dc}^{\prime} \) is the compensation for the mobile case given by Equation (53). We see that \( E_{c} \) is the sum of reduced width \( (1 - d_{hop}) \), \( E_{sc} \) and \( E_{dc} \). \( PE_{c,j} \) is the sum of \( E_{c} \) for the nodes in all the previous HCGs; \( PE_{c,j}^{\prime} \) and \( G_{HCG,j} \) respectively, are the \( PE_{c} \), \( E_{c} \) and \( G_{HCG} \) values of node \( i \) in the neighbor subset \( \{SN\} \); \( G_{\text{SMG}} \) is the gradient value computed using the simple smooth algorithm introduced in Section 2.2; and \( G_{\text{GDE}} \) is the final computed gradient value.
Algorithm 2: Compute gradient by propagation.

**Input:**
- Hop-count gradient value $G_{HCG}$ of neighbor nodes
- Gradient reduced width value $\Delta$
- Static compensation value $E_{sc}$
- Dynamic compensation value $E_{dc}$ of all previous waves

**Output:**
- GDE gradient value with varying environment parameters

**Parameters:**
- $\{SN\}$ ▶ Communication neighbor set
- $d_{hop}$ ▶ Real one hop distance
- $E_{sc}$ ▶ Compensation gradient value for the static scenario
- $E_{dc}^p$ ▶ Compensation gradient value for the dynamic scenario

$$E_c = (1 - d_{hop}) + E_{sc} + E_{dc}^p$$

for each node $i$ in $\{SN\}$ do
  if $G_{HCG,i} = (G_{HCG} - 1)$ then
    sum $(P E_{c,j} + E_{c,j})$
  end if
end for

$P E_c = \text{average}(P E_{c,j} + E_{c,j})$

$G_{GDE}^p = G_{SMG} - P E_c$

5. GDE ALGORITHM ANALYSIS

To validate our algorithms, we run simulations in Matlab. Section 4 presents the mechanisms used by the GDE algorithm to adapt to the various parameters. In this section, we test the performance of that GDE algorithm, focusing on node density, node speed, circular gradient waves, spatial node distribution, multicast percentage, communication round length, realistic model of transmission range and adaptive behavior.

The deployment area of nodes is a square of $1000 \times 1000$ m. The transmission range $r$ of each node is 80 m. The node density $n_t$ is the average number of nodes within the transmission range of each node. In the test of Section 5.3, one seed node is placed on the lower left corner of the square, and each node makes a distance estimation from itself to the seed node. This test presents the performance of GDE under the influence of circular gradient waves. In the other performance tests, the seed nodes are placed on the left border of the square, and each node makes a distance estimation from itself to the line of seed nodes. The purpose of using a line of seed nodes is to test how GDE works for each parameter in each type of test without the influence of the circular gradient waves presented in Section 4.1. The testing result is under one specific influence, and without the approximation error caused by the circular gradient waves. The test for the adaptive behavior shows the performance of the GDE algorithm with integrated parameters using Algorithm 2. For the static node deployment, nodes are scattered on the deployment field with the specific spatial node distribution according to the testing purpose. For the dynamic node deployment, each node follows the Random Walk model. We run each experiment 20 times for each testing point.

The error for each node is computed as $\varphi = (|\xi - \theta|)/r$, in which $\xi$ is the estimated distance to the line of seed nodes using different algorithms, $\theta$ is the real distance toward the line of seed nodes, $r$ is the transmission range of each node and $\varphi$ is the error normalized to the transmission range $r$. The HCG and SMG algorithms compute the gradient values $G_{HCG}$ and $G_{SMG}$, and the estimated distance is computed by $\xi_{HCG} = G_{HCG} \cdot d_{hop}$ and $\xi_{SMG} = G_{SMG} \cdot d_{hop}$, respectively. The GDE algorithm uses the mechanisms presented in the previous sections to compute the gradient value $G_{GDE}$, and the estimated distance is computed by $\xi_{GDE} = G_{GDE} \cdot r$.

5.1. Node density

With the first set of tests we want to find what are the effects of a varying node density and network diameter on the accuracy of the algorithms. All nodes are static and uniformly distributed, and broadcast locally to their neighbors.

The qualitative analysis on the average absolute error of the distance estimation of HCG, SMG and GDE algorithms is presented in Fig. 12a for various node densities and network diameters. On the $x$-axis we see the node density and on the $y$-axis the network diameter. On the $z$-axis we show the average absolute error. The results show that the error stops decreasing for HCG for a node density larger than 15. The distance estimation errors using SMG and GDE algorithms decrease when the node density increases. The error of GDE is always smaller than the one of SMG. When node density is constant, and the network diameter increases, the errors of both SMG and GDE have a small increase but only by a small factor. This is mainly caused by the cumulative effect of the distance estimation error of the nodes in the previous hop-count waves. Overall, the error of GDE is smaller than the errors of both HCG and SMG.

The quantitative analysis of the average absolute error of the three algorithms is presented in Fig. 12b for various node densities. The network diameter is 12. The GDE method reduces the average error of HCG and SMG significantly for node density higher than 15. The standard deviation of the error for both SMG and GDE decrease when the average node density increases. The standard deviation of the estimation error for GDE is always smaller than the one of SMG for node density higher than 15. For an average node density of 30, the average estimation errors for HCG, SMG and GDE are 0.31, 0.24 and 0.15. While the standard deviations are 0.025, 0.036 and 0.020, respectively. Figure 12b also shows the error reference value according to Equation (36). As the node density increases, the error of the GDE algorithm converges toward the reference error value.

5.2. Node speed

The second set of experiments is intended to find the effects of various node speed levels and node densities on the accuracy of
the algorithms. Nodes are uniformly distributed, and broadcast to their neighbors. They move following the Random Walk model, and the network diameter is 12.

The qualitative analysis on the average absolute error of the distance estimation of HCG, SMG and GDE algorithms is presented in Fig. 13a for various node speeds and node densities. The x-axis shows the node density, the y-axis the node speed and the z-axis the average absolute error. On the y-axis, the speed is increased in steps of 2 m/s. The estimation error of GDE decreases as node density increases. On the other hand, it increases slightly when the average node speed increases. Also, when the node speed increases, the errors of HCG and SMG also increase. For node speeds higher than 8 m/s, the error of SMG increases greatly and is even larger than the one of HCG. This is mainly caused by the fact that SMG is designed for the static environment and it does not consider the influence of mobile nodes.

The quantitative analysis of the average absolute error of the three algorithms is presented in Fig. 13b for various node speeds. The node density is set to 50. The network diameter is 12. As the speed increases, the error of GDE increases slightly, but SMG increases significantly. When the node speed is 20 m/s, the average error of distance estimation computed by GDE is 0.148, while the error levels for HCG and SMG are 0.391 and 0.610, respectively. The standard deviation for HCG, SMG and GDE are 0.011, 0.019 and 0.010, respectively. The test results show that the GDE algorithm has a good performance in a dynamic environment. HCG and SMG are not suitable in a dynamic environment, especially at high node speed.

We also perform tests for a more realistic mobility model, SLAW [12]. Figure 14 shows the error of the three algorithms for various node densities. In the experiment, all nodes are uniformly distributed in the area, the movement distance per second is \( \frac{1}{40} \) of the transmission radius. GDE shows the best performance with a node density of 15. The error of GDE decreases from 0.286 of node density 20 to 0.092 of node density 50. The standard deviation of GDE generally is also smaller than HCG and SMG. The test results show that the GDE algorithm can also work well in a dynamic environment with the SLAW movement model.
5.3. Circular gradient waves

In the real application, usually only a small number of seed nodes can be deployed in the area. In such a condition, for each seed node, the gradient created becomes a circle. As shown in Fig. 15a, a seed node is fixed at the position (0, 0). The gradient shape becomes circular arcs instead of parallel lines. In this section, we test whether the GDE algorithm exhibits good performance given that there is only one seed node. At the same time, as shown in the Section 4.1, Algorithm 1 provides a method to take advantage of the characteristic of circular gradient waves. We also want to know whether Algorithm 1 can improve the accuracy of the GDE algorithm.

In this experiment, only one seed node is fixed at position (0, 0). All the nodes are uniformly distributed and move following a Random Walk model, and the traveled distance per second is \( \frac{1}{20} \) of the transmission radius \( r \). Figure 15b shows the testing result. As the node density increases from 15 to 50, the error of GDE decreases from 0.315 to 0.132. HCG has almost no improvement as the node density increases, and SMG only improves from 0.514 to 0.159. At the same time, \( R-GDE \), which represents the GDE algorithm using Algorithm 1 in the Section 4.1, has the best result accuracy from a node density of 10–50. This means that Algorithm 1 in the Section 4.1 can increase the accuracy of distance estimation for the circle gradient shape, although the improvement is not that large. Therefore, GDE can work well even if there are only a small number of seed nodes. Thus, the GDE algorithm provides good distance estimation in both scenarios—for using a line of seed nodes and for using only one seed node.

5.4. Spatial node distribution

In this section, we check the influence of different spatial node distribution on the accuracy of the algorithms. Nodes follow again a Random Walk mobility model with speed of 2 m/s. The seed nodes are placed on the left border of the square. This time the nodes are not uniformly distributed. Instead, as shown in Fig. 16a, generally there are four fields with higher node density. The centers of the four fields are (0.25, 0.25), (0.75, 0.25), (0.75, 0.25), (0.75, 0.75), in which all the coordinates are normalized from 1000 m to 1 unit. The node distribution follows a Normal distribution with standard deviation 0.1. The GDE gradient is computed using the algorithm shown in Section 4.6.

Figure 16b shows the testing results. The \( x \)-axis is the distance to the line of seed nodes, \( y \)-axis is the average distance estimation error. The errors of HCG and SMG increase significantly from the place that is closer to the seed line to the place that is the most distant from the seed line. For GDE, the average error of distance estimation is quite low compared with the other two algorithms. At a distance of 800 m, the error is only 0.196 of the transmission range. It can also be noted that generally there are two peak values in the results of the GDE algorithm. This is mainly because, along the dimension from the

---

**FIGURE 14.** The absolute distance estimation error for various node densities with SLAW movement model.

**FIGURE 15.** (a) HCG waves with one seed node at position (0, 0). (b) The absolute distance estimation error for various node densities.
place that is closer to the seed line to the place that is further from the seed line, there are two fields with Normal node distribution. So GDE shows good performance given non-uniform spatial node distributions.

5.5. Multicast percentage

For wireless systems where nodes follow duty-cycled sleep schedules, nodes that transmit can only be heard by a subset of their neighbors. The percentage of nodes that take part determines the speed of information diffusion. For this experiment, nodes are uniformly distributed with an average density of 50. The network diameter is 12. The nodes take a Random Walk with a speed of 2 m/s.

In Fig. 17a, the x coordinate shows the multicast percentage, while the y coordinate shows the error level. Each node randomly selects a subset of its neighbors at each time step. The multicast percentage varies from 0.1 to 1.0. The error of both SMG and GDE algorithms decreases significantly. GDE always computes a better estimate than SMG. This is shown by both the average value and the standard deviation. The estimation error of HCG decreases for the multicast percentage between 0.1 and 0.5 and shows almost no improvement afterward. For an average multicast percentage of 0.6, the error and the standard deviation of GDE are 0.122 and 0.009, respectively, and the values are almost the same as the results of Section 5.1 with node density 30. This experiment shows that, even when nodes are not able to talk to all their neighbors, the GDE shows good performance.

5.6. Communication round length

The GDE algorithm uses a discrete time model, with unsynchronized communication rounds of size $P$ (each node is allowed to perform a single action during one communication round). In this section, we test the influence of the length of the
communication round on the accuracy of distance estimation. For this experiment, nodes are uniformly distributed, with an average node density of 50. They use broadcasting to communicate with their neighbors, and the network diameter is 12. The nodes move with Random Walk with a speed of 2 m/s.

In Fig. 17b, the $x$-axis is the round length in seconds, while the $y$-axis is the estimation error. As the round length increases, the error and standard deviation of SMG increases significantly. On the other hand, the error and standard deviation of GDE shows a much smaller increase from 0.097 and 0.006 to 0.113 and 0.014. The test results show that the GDE algorithm offers good distance estimation without requiring very frequent communication with neighboring nodes.

5.7. Realistic model of transmission range

In the theoretical analysis, we have assumed that the transmission range of the node is a circular disk. For the real application, the shape of the transmission area depends on many factors, and the shape can hardly be a circle. In Section 4.5, we presented a method to compensate the influence of the transmission range of the node that is not a circular disk. To test our algorithm, we make a simple transmission range model, which assumes that the transmission distance of the node changes according to a Normal distribution. The transmission circle of the node is sliced into 36 pieces evenly. For each slice, the transmission distance follows a Normal distribution. The mean value equals the theoretical transmission range $r$. The maximum transmission range is limited to twice the length of the transmission range. Figure 18 shows a model of the transmission range of the node, in which the blue circle is the expected circle-shape transmission range, and the red shape is the realistic model of the transmission range.

In the experiment, the transmission distance of each range slice for every node changes at every second. The standard deviation is normalized to $\frac{1}{30}$ of the transmission range. All the nodes are uniformly distributed with the Random Walk model, and the movement distance per second is $\frac{1}{40}$ of the transmission radius. Figure 19a shows the relation between the error of the distance estimation, the node density and the value of the standard deviation. As the standard deviation increases from $\frac{1}{30}$ of the transmission range to $\frac{1}{2}$ of the transmission range, the errors of HCG and SMG increase significantly. And the error of SMG is even larger than HCG. The GDE algorithm generally keeps a low error of distance estimation, but still as the standard deviation increases the error also increases.

Figure 19b shows the performance of GDE algorithm with standard deviation of $\frac{1}{10}$ the transmission range. From node density 20–50, the error of GDE decreases from 0.250 to 0.107.
At the same time, it keeps a small standard deviation from 0.032 to 0.012. The results show that the GDE algorithm can work in the condition that the shape of the transmission range is not a circle.

5.8. Adaptive behavior

In the experiments of the previous sections, we tested the GDE algorithm under the influence of different environmental parameters. In real deployments though, nodes are not uniformly distributed, node speed varies, the local interaction is described by a multicast communication, etc., which means the affects of all these parameters will show on the GDE algorithm at the same time. In this section, we test the GDE algorithm with all the parameters together.

In this experiment, the node distribution is the same as the experiment in the Section 5.4, which generally has four high node density fields in the area. The mobility model is still Random Walk and node speed follows a Normal distribution with mean value of 2 m/s. The multicast percentage depends on the local density. The number of successfully communicated neighbors is within a minimum of 15 nodes to a maximum of 30 nodes. For each neighbor, the probability to achieve successful communication is given by \( p = (0.2 \cdot n_t + 15)/n_t + q \), \((15 \leq n_t \leq 30)\), in which \( n_t \) is the number of nodes in the local transmission range and \( q \) is a Normal distribution factor with \( N(0, 0.1) \). The factor \( q \) simulates the random changes in the communication, which randomly affects the successful rate of the communication. Therefore, the variable of the number of successful transmissions generally follows a Binomial distribution. The communication round length changes according to the local node density and also follows a Normal distribution, with \( N(m, 0.5) \), in which \( m \) is the mean value of the communication round length, \( m = (30 - (0.2 \cdot n_t + 15))/15 \cdot 3 \). At the same time, this experiment requires that if a node is in the sleeping period of the communication round, it cannot communicate with its neighbors and does not acknowledge messages, even if some other nodes select it as the communication target. The shape of the transmission range of each node changes for every second, which is the same as in the Section 5.7.

Figure 20 shows the experiment results. As the distance to the seed nodes increases, the distance estimation error also increases. The GDE algorithm has the best performance, and the estimation error grows from 0.200 for 100 m to 0.868 for 800 m. Generally, according to the two experiments, GDE has basic adaptive capabilities, and can produce good distance estimation results in all these scenarios. The simulation results complement the analysis in the previous sections and show that the GDE algorithm is suitable and applicable to realistic application scenarios.

6. APPLICATIONS OF GDE

Many applications need accurate distance estimation. In this section, we provide three kinds of applications where the GDE algorithm can be used: cluster center detection, overlay shape construction and routing. Generally, for these applications, the gradient that the GDE algorithm builds may provide direction information as well.

6.1. Cluster center detection

Some clustering algorithms such as ASH [13] can create clusters of nodes in large-scale networks. One question is how to find the node that is closest to the center of the cluster. The gradient created by the GDE algorithm provides a solution to finding the center of a cluster.

In this experiment, all nodes in the area are organized into two different groups. The first group is the square in the center of the whole area. All the nodes in the square field form a cluster. The second group is made out of nodes that are not in the square. Figure 21a shows the two groups. For this application, we need to find the node that is closest to the center of the cluster. The center of the cluster is defined as the average value of the coordinate vectors of all nodes in the specified cluster.

In the initialization step, each node knows the cluster it belongs to. If a node can discover some nodes in a different cluster, then it becomes a seed node. Each node uses the gradient creation algorithms presented in the previous sections. The nodes with the lowest gradient value are selected as the center of the cluster.

Figure 21b–d show the selected center nodes, using HCG, SMG and GDE algorithms, respectively. In the picture, the bigger nodes are the selected center nodes, and the green cross mark is the expected real center.
To evaluate the performance of cluster center detection by different algorithms, we set two kinds of statistical indicators. The first one is the average distance from the selected center nodes to the real center of the cluster. The second one is the number of selected center nodes. In the experiments, the nodes are uniformly distributed in a square area. Therefore, the average value of the coordinate vectors of all nodes can represent the center point of the distributed nodes in the space. If the nodes are not uniformly distributed in the area, the above two evaluation metrics can also be used. But the center of the cluster should be recalculated according to the requirements and spatial node distribution, which is beyond the research scope of this paper.

The side length of the square is normalized to 1. The cluster field is the square with a length of 0.8 in the center of the whole area as shown in Fig. 21a. The transmission radius of the node is 0.08. The evaluations are made in both static and dynamic environments. In the dynamic testing, the nodes follow a Random Walk model and the movement distance per second is 0.002.

Figure 22a shows the average distance to the real center of the cluster using three algorithms. For node density 15–25, SMG has the best accuracy. And from node density 30–50, GDE has the best accuracy. At node density 45 and 50 testing point, the error of HCG is even smaller than SMG.

Figure 22b shows the number of center nodes selected by the algorithms. SMG and GDE always select \( \leq 2 \) nodes as the center nodes, while HCG selects more than 10 nodes as the center nodes. In the dynamic environment, the testing results...
are shown in Fig. 23a and b. Still GDE offers the best accuracy and selects only one or two nodes as the center nodes.

The reason why HCG gives the worst accuracy and selects many nodes as the center nodes is because of its low resolution. The nodes in the same HCG wave take the same gradient value; then there are many nodes taking the same highest gradient value. We see that SMG and GDE both have higher resolutions. So both algorithms have better results. Although GDE is based on the SMG created by SMG, GDE makes many compensation computations, which in advance smoothed the gradient of SMG. Therefore, from node density 30, GDE has much better accuracy than SMG in both static and dynamic situations. Because the node density cannot be infinite, there is a high probability that there is no node at the position that is computed as the center. So, even using the GDE algorithm, there may be more than one nodes that are selected as the center nodes.

According to the simulation results, GDE can be used in cluster center detection with high accuracy to the center of the cluster and low number of selected center nodes.

### 6.2. Overlay shape construction

Since using the distance estimation algorithm GDE, nodes can compute the distance to some fixed nodes with known positions, then the estimated distances to more than one fixed positions create a coordinate system. According to the coordinate values, each node in the network can be marked with required colors, and all the colored nodes can construct a specified overlay shape. The application of a programmable self-assembly strategy is shown in [14] where autonomous nodes assemble into global shapes, for which the GDE algorithm can also be used. This section presents the process of creating an overlay shape using distance estimation algorithms.

#### 6.2.1. Construction with two seed lines

In the experiment, we suppose all nodes are uniformly distributed in a square area. To create a coordinate system, two lines of seed nodes are set vertically and horizontally at the border of the square. Each node computes the distances to the
two lines; then the two distances can be used as the coordination values. Using the created coordinate system, any overlay shape can be constructed. Figure 24 shows the node distribution on the deployment area and the double circle shape to be constructed.

To evaluate the performance of different algorithms to construct a specified shape, we use two evaluation metrics. The first metric is the number of nodes outside the specified shape. If the real position of the node is not inside the shape, but the node is computed as in the shape, or if the real position of the node is inside the shape, but the node is not computed as in the shape, then it is counted as an error node. We count the total number of error nodes. The second metric is the average distance of all error nodes to the specified shape. The distance is defined as the shortest distance from the node to the border of the shape.

At the same time, we set two rounds of tests to evaluate the absolute coordinate error and the relative coordinate error of the shape, respectively. In the first round of tests, we evaluate the absolute coordinate error of the overlay constructed shape. The center of the desired shape is specified in the center of the deployment area. The nodes construct the overlay shape based on the calculated coordinate values. Figures 25a, 26a and 27a show the constructed shape using the HCG, SMG and GDE algorithms, respectively, with the specified center. In the second round of tests, we evaluate the relative coordinate error of the overlay constructed shape. The center of the desired shape is calculated using the average vector values of all the nodes constructing the overlay shape. The constructed overlay shape is the same as the first round of tests. Figures 25b, 26b and 27b show the constructed shape calculated in the first round of tests with the recalculated center. Compared with Figs 25a,b, 26a,b and 27a,b, the recalculated centers are different from the specified center.

In the experiments, the nodes are uniformly distributed in a square area. The length of the side of the square area is normalized to 1. The standard shape is constructed by two concentric circles as shown in Fig. 24. The specified center of the shape is (0.5, 0.5). The smaller circle has the radius 0.15. The bigger circle has the radius 0.3. The transmission radius of the node is 0.08. We take the experiments in both static and dynamic environments. In dynamic testing, the nodes follow a Random Walk model and the movement distance per second is 0.003.
Figure 26. (a) Created shape using SMG with specified center of the shape. (b) Created shape using SMG with recalculated center of the shape.

Figure 27. (a) Created shape using GDE with specified center of the shape. (b) Created shape using GDE with recalculated center of the shape.

Figure 28a and b show the testing results in the static environment. The entries HCG-S, SMG-S and GDE-S are the testing results with the specified shape center, and the entries HCG-R, SMG-R and GDE-R are the testing results with the recalculated shape center. As shown in Fig. 28a, the GDE algorithm can produce the minimum number of error nodes. From node density 10 to 80, the number of error nodes of SMG-S, SMG-R, GDE-S and GDE-R increases, but HCG-S and HCG-R increase by a large number. The reason is that HCG algorithm has very low resolution, which can hardly compute the correct coordinate values of each node. The number of error nodes of the HCG and SMG algorithms decreases as the centers of the shapes are recalculated.

The errors of the HCG and SMG algorithms not only distort the shape but also offset the coordinate values. The recalibration of the shape center decreases partially the error caused by the offset to the coordinate values, which decreases the number of error nodes. But the numbers of error nodes of the GDE algorithm increase as the center of the shape is recalculated.

Because the GDE algorithm has accurately calculated the coordinate values of the nodes, recalibration of the shape center can introduce more error. So the result of GDE-S produces the most accurate results. Figure 28b shows the average error distance for different node densities. As the node density increases, the errors of all algorithms decrease because, as the node density increases, the accuracy of distance estimation of
all the algorithms will increase. The GDE algorithm holds the minimum average error distance to the specified shape.

GDE also works very well in the dynamic environment as shown by Fig. 29a and b. The variation trend of the results is almost the same as in the static environment. GDE-S holds the best testing results in both number of error nodes and average error distance. Therefore, the GDE algorithm can be used for overlay shape construction, and works well in both static and dynamic environments.

6.2.2. Construction with two seeds
In the previous discussion and testing, in order to create a coordinate system, two lines of seed nodes are placed at the borders of the area. But in real applications, it is not feasible to create a line of seed nodes. Instead, we can only use a small number of nodes as seeds. Even when using a small number of seed nodes, the coordinate system can also be created using the GDE algorithm. Using the estimated position information, the overlay shape can also be constructed.

To create the coordinate system, the minimum number of seed nodes that are needed in a square area is 2. And the two seed nodes should be placed at some known positions. Then any node and the two seed nodes can create a triangle, in which the lengths of the three edges can be computed and the positions of the two seed nodes are known. Using this triangle, the position values of the nodes can be computed.

Suppose that the side length of the square area is 1. In the experiment, we put two nodes at the corner of the square \((0, 0)\) and \((0, 1)\) as the seed nodes. Then each node computes the distances to the two seed nodes and the coordinate values. As shown in Fig. 30, node \(n\) is in the area with two seed nodes at the corner. Suppose that the position of the two seed nodes are known, and the distance of the two seed nodes are \(l\). Using the GDE algorithm, the distances from the seed nodes to node \(n\) are
Then it is easy to compute the coordinate values of node \( n \) using Equation (59):\\n
\[
\begin{align*}
\begin{cases}
x = \sqrt{d_2^2 - y^2}, \\
y = \frac{d_2^2 - d_1^2 + l^2}{2l}.
\end{cases}
\end{align*}
\]

We take the same evaluation metrics and the same testing parameters as shown in Section 6.2.1, except that there are only two seed nodes. The two seed nodes are at the corners of the square \((0, 0)\) and \((0, 1)\). Figure 31a shows the number of error nodes with specified shape center and recalculated shape center, and Fig. 31b shows the average error distance with specified shape center and recalculated shape center. As the node density increases from 10 to 80, the GDE-S value generally holds the minimum number of error nodes and the minimum average distance. The tendency of the testing results and the relative relations among different testing algorithms are the same as the testing results shown in Section 6.2.1. This means that the overlay shape can be accurately constructed and the GDE algorithm can calculate high accuracy coordinate values by using only two seed nodes.

The coordinate values of nodes can be calculated with a limited number of seed nodes. Nagpal et al. [3] analyzes the effect on the accuracy of the distance estimation caused by the positions of the seed nodes. For different seed node positions, the equation to calculate the coordinate values will be different from Equation (59), but the process of calculating the coordinate values described in this section can still be used.

In this section, we have successfully built a virtual coordinate system in the network with two seed nodes, and constructed an overlay shape based on the virtual coordinate system. According to the testing results, GDE offers the highest accuracy compared with the HCG and SMG in the overlay shape construction.

**Figure 30.** Computation of the coordinate values of node \( n \) based on two seed nodes.

**Figure 31.** The entries HCG-S, SMG-S and GDE-S are the testing results with the specified shape center, and the entries HCG-R, SMG-R and GDE-R are the testing results with the recalculated shape center. (a) The number of error nodes in the shape with two seed nodes. (b) The average error distance of error nodes in the shape with two seed nodes.

### 6.3. Routing

Accurate coordinate systems can be constructed based on the high distance estimation accuracy of the GDE algorithm. At the same time, geographic routing can also be implemented using the created coordinate system, such as [15]. Geographic routing takes advantage of location information of nodes to forward packets to specific geographic positions. Using GDE, our virtual coordinate system can be used for geographic routing algorithms without relying on location information.

Nevertheless, in this section, we do not employ geographic routing based on the virtual coordinate system, but fully take advantage of the gradient itself to forward messages, because the gradient value encodes also direction information, which can be used to find the target nodes. We present a demonstration of the ability of using the gradient values to make routing from one node to another node in a network. The routing mechanism can
be widely used as a building block in wireless sensor networks, such as [16]. For the purpose of evaluation, we use the simple routing mechanism which finds the route in a greedy manner to a destination.

Suppose that there are many nodes in the area, and for some reason there is no node in the middle of the area (e.g. in a stadium). Figure 32a shows the distribution of nodes, in which the blue node is the target node, and the red node is the source node. We use the gradient to create a route from the source node to the target node in the network. First, we set the target node as the seed node, and then a gradient can be created in the network. Any node that wants to send information to the seed node just selects the node with the largest gradient value as the next hop in its own transmission range, and forwards messages to that node. Because the seed node holds the largest gradient value in the network, the messages will finally reach the seed node. Figure 32b–d show the route from the source node to the target node using the HCG, SMG and GDE algorithms, respectively. In this experiment, the node density is 20, and all nodes are static.

Even if there is a hole in the middle of the area, the gradient can be created, and the route from the source node to the target node can be found. The route selection does not depend on any coordinate system, but on the gradient in the network. The route to the destination shows the direction characteristic in the gradient value. This is because the gradient value presents the network connectivity, and avoids the real geographic position of the hole. It can also be found that the...
routes selected by GDE and SMG have the same number of hops. We see that HCG selects a different route but the number of hops is almost the same as the hop number of GDE and SMG. This is because the routing strategy here is to find the shortest path in a greedy manner, and so the next hop node is always selected among the nodes in the HCG wave with larger HCG values. Therefore, there is not that much difference for all three algorithms to select the route. But if the routing strategy does not require the shortest path to the destination, but requires to find a specific geographic position in the network, the results of the three algorithms will be different. The GDE algorithm can provide better results than SMG and HCG, because the GDE algorithm is the more accurate than the other two algorithms in calculating the coordinate values.

The gradient distance estimation algorithms HCG, SMG and GDE can be used in the network routing. It offers much potential in the routing application which requires accurate geographic position information.

7. RELATED WORK

The research on spatial computing includes computer architectures, programming languages, spatial algorithms, etc. Movable Feast Machine (MFM) [17] explores the spatial tiling of computer architectures and their associated computational model, which shows robust and indefinite scalability. Growing Point Language (GPL) [18] from MIT amorphous computing project organizes the networks into complex spatial patterns. Proto programming language [19] controls the behavior of spatial computers by approximating the continuous space. The middleware and programming approach Tuples On The Air (TOTA) [20], based on spatial distributed tuples, is designed to program pervasive and mobile network computers. Some algorithms that can be used in spatial computing are created. For example, the ASH [13] algorithm creates a quasi-static overlay on top of a large-scale mobile network using only local interactions.

Wireless sensor networks have some similar concepts characteristics to spatial computing, and have already a large number of applications, such as environmental monitoring [21, 22], structural monitoring [23], target tracking [24, 25], surveillance missions [26], etc. Distance estimation for wireless systems is one of the most important building blocks for various services such as routing [27], clustering [28], localization [1, 26], etc.

Many clustering algorithms [29] and geographic routing algorithms [15] assume that nodes are aware of their location information. While this is a feasible assumption in some settings (e.g., nodes have GPS [30] modules attached), there are many cases where node location information is either not available such as indoors, underground, underwater, as so on [31], or it requires additional expensive equipment. The paper [27] describes algorithms that allow nodes to calculate the virtual coordinate values in the whole network without any location information. The purpose is to enable the other geographic routing algorithms to work over the constructed virtual coordinate systems. But the algorithm in the paper does not consider the real application problems, such as varying node density, node mobility, etc. And the construction of the virtual coordinates needs the movement of nodes in the network, and the results of the algorithm will be different. The GDE algorithm can provide better results than SMG and HCG, because the GDE algorithm is the more accurate than the other two algorithms in calculating the coordinate values.

The gradient distance estimation algorithms HCG, SMG and GDE can be used in the network routing. It offers much potential in the routing application which requires accurate geographic position information.
Thus they are successfully used in many applications for large-scale systems, such as routing [43].

The algorithm [3] serves as a starting point for our work and builds a self-organized, global coordinate system on top of an ad hoc wireless sensor network. The GDE algorithm relies only on distributed, simple computations and local communication. The algorithm is able to cope well with node failures or changes in network sizes.

8. CONCLUSIONS AND FUTURE WORK

Lately, a multitude of systems require the use of spatial computing, wireless sensor networks, robot swarms, amorphous computers being just a few examples. Distance estimation is of tremendous importance for obtaining spatial information and allows building blocks such as virtual coordinate systems to be employed. The majority of the algorithms that require position information presume knowledge about node position via systems such as GPS. While this approach is feasible for some application scenarios, in a lot of cases it suffers from frequent unavailability and high costs in terms of energy consumption.

For this reason, in this article, we introduce a novel distributed algorithm called GDE for the estimation of distances in spatial computing. It is based on the gossiping mechanism to estimate distances between nodes, using solely local interactions. GDE also considers the influence of real application conditions, including node mobility, non-uniform spatial node distribution, multicast percentage, communication round length, transmission range, etc. The theoretical model and the experimental evaluation by means of simulations show that GDE succeeds in estimating the distances between nodes in both static and dynamic scenarios with considerably high accuracy, even under the influence of different kinds of environmental parameters. For static network topologies with uniform distribution of nodes, GDE approaches the boundary of what is practically achievable, offering for high node densities an error estimate three times smaller than the simple HCG. Simulation results show its strengths in more difficult scenarios—the non-uniform distribution of nodes being one in which the error is reduced with a few orders of magnitude when compared with existing state of the art.

We also introduce some practical applications using the GDE algorithm: cluster center detection, overlay shape construction and routing. Testing GDE in three typical applications shows that it can be successfully used with considerably high accuracy.

The GDE algorithm is presented from a theoretical point of view, and still has many details that could be further investigated. Generally, there are four main future research directions.

First, the article already presents methods to compensate for the errors caused by various parameters. The models used in the tests are all based on theoretical conditions. So further investigation should extend the tests to more realistic deployment conditions, such as real human mobility models, real communication models, etc.

Secondly, the accuracy of the GDE algorithm could be further improved, once the specific deployment parameters are known. At the same time, it is worth investigating how other parameters, which are not analyzed in the paper, influence the performance of the GDE algorithm.

Thirdly, the influence of the real transmission range of nodes to the accuracy of GDE is an investigation point. There is a very basic foundation for the GDE algorithm, which requests the transmission range to be a circle or at least to be generally symmetric. This can only happen in an ideal environment. Section 4.5 presents a method to compensate the influence of the irregular transmission range. The mathematical model of the transmission range still assumes that the average values of the transmission distance on each direction are the same. In the real application, such as in buildings, the shape of the transmission range is sometimes unpredictable. It is even impossible to guarantee that the transmission range is generally symmetrical. How to make the distance estimation algorithm work better with irregular transmission ranges is a subject of future work.

Finally, the GDE algorithm produces higher accuracy for higher node density. In some practical applications, the network is sparse. Using multi-hop communication instead of gossiping can increase the number of communicated partners of each node, but this may also increase the message complexity and introduce other communication problems. How to make GDE work better in a low node density area could be further investigated.

FUNDING

Qingzhi Liu is supported by the SHINE (Sensing Heterogeneous Information Network Environment) project. SHINE is the flagship project of DIRECT institute, which has started in 2012 and will end in 2015. DIRECT is the virtual institute of Delft University of Technology for all research on Information and Communication Science and Technology.

REFERENCES


