Regular Article

An Online Distributed Boundary Detection and Classification Algorithm for Mobile Sensor Networks

Pham Duy Hung[1], **Tran Quang Vinh**[1], **Trung Dung Ngo**[2]

[1] University of Engineering and Technology, Hanoi, Vietnam
[2] More-Than-One Robotics Laboratory, University of Prince Edward Island, Canada

Correspondence: Pham Duy Hung, hungpd@gmail.com

Communication: received 14 March 2017, revised 22 June 2017, accepted 9 July 2017
Online publication: 30 October 2017, Digital Object Identifier: 10.21553/rev-jec.158
The associate editor coordinating the review of this article and recommending it for publication was Prof. Tran Duc Tan.

Abstract– **We present a novel online distributed boundary detection and classification algorithm in order to improve accuracy of boundary detection and classification for mobile sensor networks. This algorithm is developed by incorporating a boundary detection algorithm and our newly proposed boundary error correction algorithm. It is a fully distributed algorithm based on the geometric approach allowing to remove boundary errors without recursive process and global synchronization. Moreover, the algorithm allows mobile nodes to identify their states corresponding to their positions in network topologies, leading to self-classification of interior and exterior boundaries of network topologies. We have demonstrated effectiveness of this algorithm in both simulation and real-world experiments and proved that the accuracy of the ratio of correctly identified nodes over the total number of nodes is 100%.**

Keywords– **Distributed boundary detection, Distributed boundary classification, Mobile sensor network.**

1 Introduction

A mobile sensor network can be deployed for surveillance and reconnaissance, security and patrolling, and environmental monitoring in an unknown environment. Coverage area of a deployed sensor network is identified through the shape formed by the boundary nodes. In this paper, we focus on developing an *online distributed boundary detection and classification algorithm* enabling mobile sensor nodes to identify whether they are (inner or outer) boundary nodes of mobile sensor networks. This algorithm plays a crucial role of boundary node identification in many mobile systems, such as controlling cohesion configuration of a robotic swarm [1], monitoring dynamic obstacles in mobile sensor networks, detecting intrusion in networks [2].

In [3–10], the boundary detection algorithms were designed for sensor networks with static nodes so they do not require node locations and relative distance between nodes rather than using connectivity information in the network. In [3], the algorithm was developed for high density of static sensor networks by using a key feature of the distribution of nodes in which the nodes on the boundary have a smaller degree than the interior ones. A node can be considered as a boundary node by estimating a global distribution and a threshold degree. In [4], the deterministic boundary detection algorithm was proposed for a large-scale network of static sensors scattered in a polygonal region, e.g., a street network. The nodes only interact each other through communication without information of coordinates and geometric distances. Based on a stronger structural criterion of particular subgraph, called *m-flower*, interior and exterior boundaries are detected if their edges are in the flower boundary. The flower pattern was also used in [6] to detect the boundaries for both unit-disk graph and quasi-unit disk graph. In [5], a distributed boundary detection algorithm for wireless sensor networks (WSNs) was developed by exploiting special structures of the shortest paths to detect the existence of hole boundaries in the network. Another algorithm using three-hop neighbouring node information for detecting boundary around an obstacle can be found in [7]. In [8], the Localized Voronoi Polygons (LVP) was used to identify boundary nodes. If the LVP of a node is a finite number, the node is considered as an internal node, otherwise, the node is set as a boundary node. In [9], the concept of independent set from the graph theory was exploited to develop an algorithm for detecting boundaries and holes in the WSNs. The algorithm includes three main steps: each node builds an one-hop graph based on connectivity information of its neighbours; independent sets of cardinality are established, and the independent sets are connected to search for closed paths. In [10], boundary nodes are identified based on computational geometry using absolute angles of their neighbours in the local Cartesian coordinate system.

For networks of mobile sensor nodes, so-called mobile sensor networks (MSNs), the cyclic-shape algorithm [11] can be considered as the first boundary detection algorithm. This method requires local network geometry in which each node checks its missing sectors to identify whether it is a boundary node.

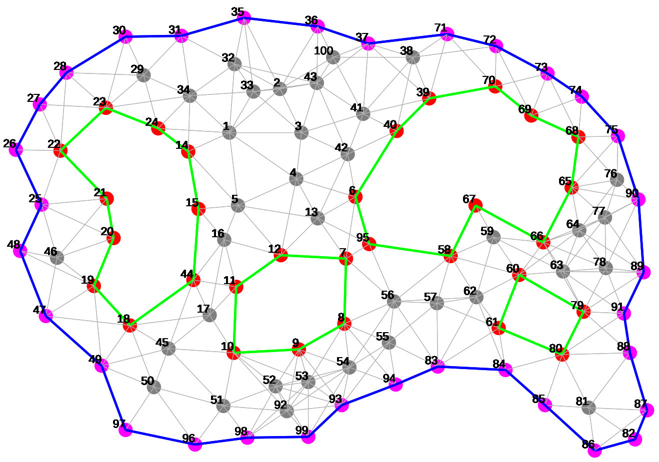


Figure 1. A demonstration of the local boundary detection and classification for a network with 100 nodes. The network is bounded by an exterior boundary (blue polygon) established by outer boundary nodes (magenta) and boundary edges (blue). Four voids inside the network are bounded by interior boundaries (green polygons) formed by inner boundary nodes (red) and boundary edges (green).

However, it exists bugs of the boundary detection algorithm because the algorithm is not purely distributed when using a recursive method with the global synchronization for resetting the suppressed error status. In another approach, each node of MSNs uses only communication to collect information from its three-hop neighbours to construct its own contours, so it does not need knowledge of relative distance between nodes [12]. Thanks to status of contours, a node is capable of identifying itself whether it is a boundary node. In [13], authors extended the idea of isocontour and proposed a boundary detection algorithm using two-hop neighbours to build its isocontour.

In this paper, we consider the boundary detection and classification problem for MSNs. The shortcoming of boundary detection errors in [11] motivates our research on developing a fully distributed boundary detection and classification algorithm based on a geometric topology approach. This paper is a comprehensive extension of our earlier work in [14]. In [14], we presented an improved boundary detection algorithm from the original algorithm in [11] by adding boundary error correction algorithm (BEC), allowing nodes to self-identify their states as either interior nodes or boundary nodes. In this paper, we have further developed this algorithm with the boundary classification (BC) algorithm to let boundary nodes self-classify as outer or inner boundary nodes, leading to self-classification of interior or exterior boundaries of mobile sensor network topologies. Last but not least, we have examined and evaluated this online distributed boundary detection and classification in both simulations and real-world experiments to prove that the accuracy of the algorithm is 100%.

The paper is organised as follows: Section 2 addresses a boundary detection algorithm. The boundary classification algorithm is described in Section 3. Experiment results are presented and discussed in Section 4. Section 5 draws conclusion.

2 Boundary Detection

Given a mobile sensor network consisting of N nodes in which each node is a mobile sensor node capable of measuring relative distance of its neighbouring nodes inside its limited sensing range r_c and carrying out peer-to-peer communication with nodes inside its communication range. In reality, the communication range is greater than the sensing range, so we assume that if two nodes are inside sensing range of each other, they can communicate to exchange information.

Geometric configuration of a mobile sensor network depends on spatial distribution of the nodes, which is considered in 2-dimensional space. The outer shape of the geometric configuration is defined as its boundary. Nodes on the boundary are called as *boundary nodes* while the others are called as *interior nodes*. Boundaries can be classified into *exterior boundaries* or *interior boundaries* in which a void inside the configuration is bounded by an interior boundary. Nodes inside sensing range of node i are called its neighbours, N_i. A connectivity between nodes i and j is denoted by e_{ij}, which is positive if $j \in N_i$ or zero if $j \notin N_i$.

2.1 Boundary Detection Method

A *local boundary detection algorithm* presented in [11] allows node i to identify itself as either a boundary node or an interior node. Node i has neighbours N_i sorted in a cyclic order. Each pair of two adjacent neighbours (j, k) make a sector with an angle θ_i^{jk} classified into two types of sector as follows:

- *Missing Sector*: A sector (j, k) is called a missing sector if nodes i and j are not connected, $e_{jk} = 0$, or if the angle between them is greater than or equal to π, $\theta_i^{jk} \geq \pi$. The angle θ_i^{jk} is called as *missing sector angle*.
- *Triangle sector*: A sector (j, k) is called a triangle sector if nodes i and j are connected, $e_{jk} > 0$. In this case, nodes i, j, k form a triangle.

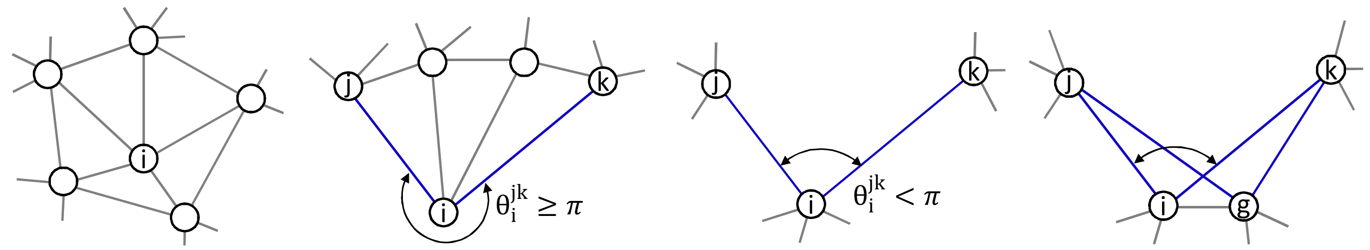
A set of consecutive sectors in node i's cyclic order is denoted by P_i. Node i uses states of sectors in P_i to classify itself as follows:

- *Interior node*: A node is self-labeled as an interior node if its all sectors are triangle sectors.
- *Boundary node*: A node is self-labeled as a boundary node if it has at least one missing sector. The edges on the boundary node's missing sector are called as boundary edges.

Because a boundary is a closed-loop interconnectivity established by boundary edges in which each boundary node is connected with two neighbouring ones, the algorithm to identify *boundary errors* is illustrated in Figure 3 and defined as follows:

Definition 1 (Boundary error)**.** *A boundary node is seen as a* boundary error *if it is directly or indirectly connected with an interior node through boundary edges.*

In other words, boundary node i with a missing sector $(j, k) \in P_i$ becomes a boundary error if at least



(a) Interior node (b) Boundary node with $\theta_i^{jk} \geq \pi$ (c) Boundary node with $\theta_i^{jk} < \pi$ (d) Both i and g are interior nodes

Figure 2. Local boundary detection algorithm

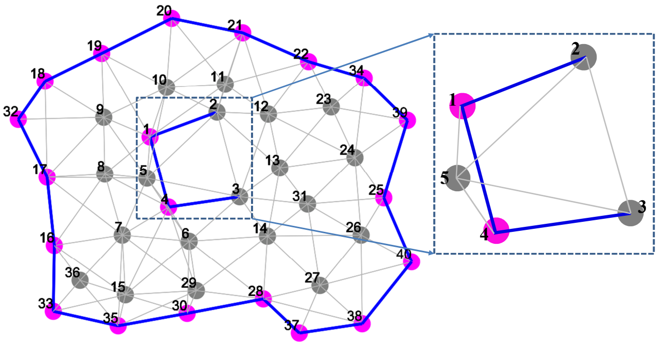
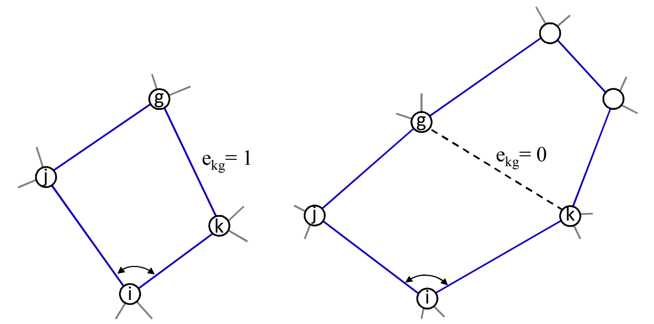


Figure 3. Boundary errors: a network of 40 nodes with some boundary errors (left), and a zoomed-in boundary error (right).



(a) Closed quadrangle (b) Open quadrangle

Figure 4. A quadrangle is established by a missing sector (j, k) and a node $g \in N_{j\setminus i}$ in two cases: (a) a closed quadrangle with $e_{kg} = 1$, and (b) a open quadrangle with $e_{kg} = 0$.

one node of the pair (j, k) or another node on the boundary indirectly connected with node i is an interior node.

The boundary errors can be removed by recursively identifying all the single-node errors and labelling them as interior nodes as described in [11]. This process requires some sort of global synchronization to periodically reset the suppressed error status on the nodes and is repeated until no existing boundary error, thus the algorithm developed in [11] is not fully distributed. Moreover, the authors claimed that the accuracy of the local boundary detection algorithm, the ratio of correctly detected nodes to the total number of nodes, reaches 86% in a static configuration in simulation. Last but not least, this algorithm has not been examined through real-world experiments.

Our motivation is to upgrade the local boundary detection algorithm to become a fully distributed algorithm, increase its accuracy of boundary detection and classification algorithm, and examine it through real-world experiments. Moreover, our algorithm in 2.2 allows boundary errors to self-correct without using the recursive process and global synchronization.

2.2 Boundary Error Correction

In this section, we present a *boundary error correction algorithm* (BEC) used to detect and remove boundary errors. It is easy to realize that a missing sector (j, k) of node i is formed in a topology with at least four edges as illustrated in Figure 4. A quadrangle is established by the missing sector and a node g in $N_{j\setminus i} = N_j \setminus \{N_i \cup i\}$ and/or $N_{k\setminus i} = N_k \setminus \{N_i \cup i\}$ formed in either a *closed quadrangle* with four edges or an *open quadrangle* with more than four edges. We release a new concept as follows:

Definition 2 (e-triangle sector)**.** *If node i's quadrangle is divided into triangles, node i's missing sector formed in the quadrangle is called an* e-triangle sector.

Intuitively, the quadrangle established by nodes i, j, k, g can be divided into triangles by one of three cases as follows:

- Existing node j's triangle sector $(g, m) \in P_j$ as illustrated in Figure 6.a
- Existing node k's triangle sector $(g, \ell) \in P_k$ as illustrated in Figure 6.b.
- Existing node g's triangle sector $(\ell, m) \in P_g$ as illustrated in Figure 6.c.

Let N_{ij} and N_{ik} be neighbouring nodes of node i in areas $\mathbf{S}_{ij}$ and $\mathbf{S}_{ik}$, respectively, as described in Figure 5. Conditions corresponding to the scenarios to identify whether the missing sector $(j, k) \in P_i$ is an e-triangle sector or not are given as follows:

- There exists node $m \in N_{ik}$ connected to node j, $e_{jm} = 1$, so that nodes j, m have at least one common neighbour g not in N_i, $N_{j\setminus i} \cap N_{m\setminus i} \neq \emptyset$.

$$cond_1 = \begin{cases} 1 & \exists m \in N_{ik} : (e_{jm} = 1) \wedge (N_{j\setminus i} \cap N_{m\setminus i} \neq \emptyset) \\ 0 & \text{Otherwise} \end{cases} \tag{1}$$

The pair (i, g) of node j contain two triangle sectors $(i, m) \in P_j$ and $(g, m) \in P_j$, so it is not a missing sector. Thus, node j is not a boundary node of the quadrangle; that is, it is an interior node.

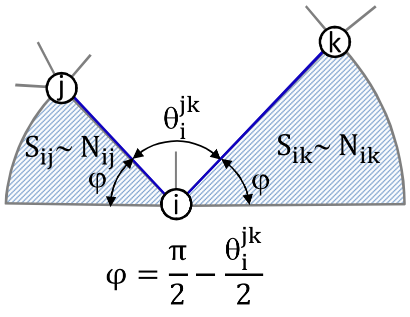


Figure 5. Subset of neighbouring nodes

- There exists node $\ell \in N_{ij}$ connected to node k, $e_{k\ell} = 1$, so that nodes k, ℓ have at least one common neighbour g not in N_i, $N_{k\backslash i} \cap N_{\ell\backslash i} \neq \emptyset$.

$$cond_2 = \begin{cases} 1 & \exists \ell \in N_{ij} : (e_{k\ell} = 1) \wedge (N_{k\backslash i} \cap N_{\ell\backslash i} \neq \emptyset) \\ 0 & \text{Otherwise} \end{cases} \tag{2}$$

The pair (i, g) of node k contain two triangle sectors $(i, \ell) \in P_k$ and $(g, \ell) \in P_k$, so it is not a missing sector. Thus, node k is not a boundary node of the quadrangle; that is, it is an interior node.

- There exists nodes $\ell \in N_{ij}$ and $m \in N_{ik}$ connected together, $e_{\ell k} = 1$, so that they have at least one common neighbour g not in N_i, $N_{\ell\backslash i} \cap N_{m\backslash i} \neq \emptyset$.

$$cond_3 = \begin{cases} 1 & \exists \ell \in N_{ij}, m \in N_{ik} : (e_{\ell m} = 1) \wedge \\ & (N_{\ell\backslash i} \cap N_{m\backslash i} \neq \emptyset) \\ 0 & \text{Otherwise} \end{cases} \tag{3}$$

The pair (j, k) of node g contain at least two triangle sectors $(j, \ell) \in P_g$ and $(m, \ell) \in P_g$, thus it is not a missing sector. Therefore, node g is not a boundary node of the quadrangle; that is, it is an interior node.

Proposition 1 (Boundary error)**.** *A boundary error appears on a boundary node if and only if it has an e-triangle sector.*

Proof: If node i has an e-triangle sector $(j, k) \in P_i$, it satisfies that $(cond_1 = 1) \vee (cond_2 = 1) \vee (cond_3 = 1)$. It means that one of three nodes j, k, g is an interior node. According to the definition 1, node i is a boundary error.

If a boundary error appears on a missing sector $(j, k) \in P_i$ formed in a quadrangle $\{i, j, k, g\}$, one of three nodes j, k, g must be not a boundary node of the quadrangle. Thus, one of pairs (i, g) of node j, (i, g) of node k, and (ℓ, m) of node g is not a missing sector; that is, it satisfies the one of the conditions for an e-triangle sector, $(cond_1 = 1) \vee (cond_2 = 1) \vee (cond_3 = 1)$. Hence, the missing sector $(j, k) \in P_i$ is an e-triangle sector. This completes the proof.

Proposition 2 (Boundary error correction)**.** *If all the nodes with e-triangle sectors are labeled as interior nodes, there does not exist any boundary error in the network.*

Proof: We use proof by contradiction. Assume that all the nodes with e-triangle sectors are labeled as interior nodes but there still exists a boundary error in the network. According to the proposition 1, the boundary error has an e-triangle sector. This is contradict to the assumption so the proposition is proven.

Algorithm 1: e-triangle sector detection

Input: $(j, k) \in P_i, N_i, N_j, N_k$
Output: *etrianglesector*
Initialization: $Cond_1 = 0, Cond_2 = 0, Cond_3 = 0$
etrianglesector $\leftarrow 0$
$[N_{ij}, N_{ik}] = AdjacentNeighbours(\theta_i^{jk}, \frac{\pi}{2})$
for $\ell \in N_{ij}$ **do**
  for $m \in N_{ik}$ **do**
    Get N_ℓ, N_m
    if $(e_{jm} = 1) \wedge (N_{j\backslash i} \cap N_{m\backslash i} \neq \emptyset)$ **then**
      $Cond_1 = 1$
    if $(e_{k\ell} = 1) \wedge (N_{k\backslash i} \cap N_{\ell\backslash i} \neq \emptyset)$ **then**
      $Cond_2 = 1$
    if $(e_{\ell m} = 1) \wedge (N_{\ell\backslash i} \cap N_{m\backslash i} \neq \emptyset)$ **then**
      $Cond_3 = 1$
if $(Cond_1 = 1) \vee (Cond_2 = 1) \vee (Cond_3 = 1)$ **then**
  $etrianglesector = 1$

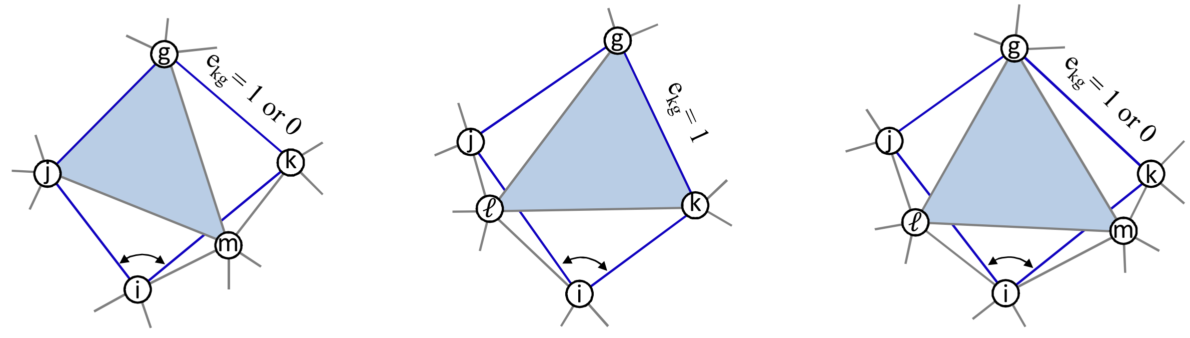
The core of the BEC is an e-triangle sector detection algorithm as shown in Algorithm 1. The algorithm considers a missing sector $(j, k) \in P_i$. It uses a function $AdjacentNeigbhours(\theta_i^{jk}, \frac{\pi}{2})$ to get node i's neighbours N_{ij} and N_{ik} in its areas $\mathbf{S}_{ij}$ and $\mathbf{S}_{ik}$, respectively. In this algorithm, all existing edges $\{e_{jm}, e_{k\ell}, e_{\ell m}\}$ on N_{ij} and N_{ik} are searched and checked whether the conditions of an e-triangle sector is satisfied. If $(Cond_1 = 1) \vee (Cond_2 = 1) \vee (Cond_3 = 1)$, it returns $etrianglesector = 1$, confirming that the missing sector $(j, k) \in P_i$ is an e-triangle sector.

This algorithm only uses sensing to identify sets N_i and P_i, and one-hop communication between node i and its neighbours j, k, ℓ, m to determine sets N_j, N_k, N_ℓ, and N_m respectively, so it is fully distributed in meaning that each node only uses local information from itself and its neighbours to classify itself as a boundary node or an interior node. The algorithm does not require global synchronization to periodically reset the suppressed error status on the nodes. Therefore, computational complexity of this algorithm is $O(N_i^2)$, but only $O(1)$ bits/node/round of communication is used to detect an e-triangle sector.

3 Boundary Classification

In this section, we present a *boundary classification algorithm* (BC) allowing a node to classify itself whether it is on an interior or exterior boundary. Figure 1 illustrates a result of this algorithm applied for a network of 100 nodes. Before presenting the algorithm, we remain the concepts of a boundary as follows:

Definition 3 (Boundary)**.** *A boundary is a closed-loop inter-connectivity containing a sequence of boundary nodes*



(a) node j is an interior node (b) node k is an interior node (c) node g is an interior node

Figure 6. Scenarios for an e-triangle sector $(j, k) \in P_i$ where node i's quadrangle is divided by a triangle sector: (a) $(m, g) \in P_j$; (b) $(\ell, g) \in P_k$; (c) $(\ell, g) \in P_g$.

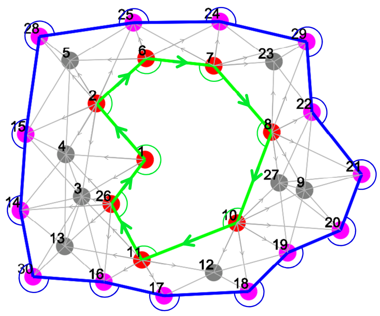


Figure 7. Boundary classification algorithm: A node $i = 1$ has a missing sector $(2, 26)$. The message broadcasted by node i is forwarded from node 2 to node 26 via the boundary edges (green). Only boundary nodes (red) $\{2, 6, 7, 8, 10, 11, 26\}$ on the green polygon are allowed to forward the message.

such that each node is connected with two neighbouring ones though two boundary edges.

Given a boundary established by n_b boundary nodes and denote Θ as the sum of the missing sector angles on the boundary. As mentioned above, there are two typical boundaries: interior and exterior boundary.

- *Interior boundary*: An interior boundary bounds a void inside the geometric configuration of a mobile sensor network, so its missing sector angles are interior angles of the polygon established by the boundary. According to polygon theorem, the sum of the interior angles of the polygon with n_b edges equals to $(n_b - 2) * \pi$. Hence, the interior boundary has property: $\Theta = (n_b - 2) * \pi$. Nodes on an interior boundary are called *inner boundary nodes*.
- *Exterior boundary*: An exterior boundary is outer shape of the geometric configuration, so its missing sector angles are complement of interior angles of the polygon established by the boundary. Hence, the exterior boundary has property: $\Theta = n_b * 2\pi - (n_b - 2) * \pi = (n_b + 2) * \pi$. Nodes on an exterior boundary are called *outer boundary nodes*.

In summary, a boundary can be classified as an interior boundary if it has $\Theta = (n_b - 2) * \pi$ or an exterior boundary if it has $\Theta = (n_b + 2) * \pi$. Any boundary node can be classified as either an inner or outer boundary node if it knows Θ and n_b. On a node, the pair (Θ, n_b) can be obtained by sending a "ping" message from a side of its missing sector to another side through a set of boundary nodes as illustrated in Figure 7. The procedure can be described as follows. In order to classify whether boundary node i with a missing sector (j, k) is an inner or outer one, node i falls into *active mode* using multi-hop communication to gather missing sector angles Θ and number of boundary nodes n_b on the boundary. It broadcasts a message, containing Θ and n_b with initialization $\Theta = \theta_i^{jk}$ and $n_b = 1$, respectively, for path discovery from node j to node k through the ad-hoc network of the mobile nodes. A node broadcasting the message is called a transmitter. All neighbours of a transmitter can receive the message, but a boundary node on its common boundary edge is allowed to forward the message with updated variables Θ and n_b. Updating variables is performed by adding the missing sector angle of the transmitter to Θ and increasing n_b by one. Thank to the message forwarded to reach node k, node i has possibility of identifying the sequence of boundary nodes on the path and checking the boundary classification condition by values Θ and n_b. Node i is self-labeled as an inner boundary node if $\Theta = (n_b - 2) * \pi$ or an outer boundary node if $\Theta = (n_b + 2) * \pi$, and rebroadcasts a message containing the list through the path in order to update the classification states for all boundary nodes on the list. When a boundary node has been self-labeled, it works in *passive mode* while the self-classification is happening on other boundary nodes. Note that, a node belongs to some difference boundaries if it has some missing sectors.

Unlike to the algorithm in [11], in our algorithm, every boundary node can be considered as a root of the boundary for self-classification when it is activated in the *active mode*. The boundary classification algorithm uses multi-hop communication, so it requests the communication complexity $O(n_b)$, where n_b is number of boundary nodes on the boundary.

4 Experiments and Discussions

4.1 Simulation

Assume that each node has ability of identifying its nearest neighbours and measuring their relative

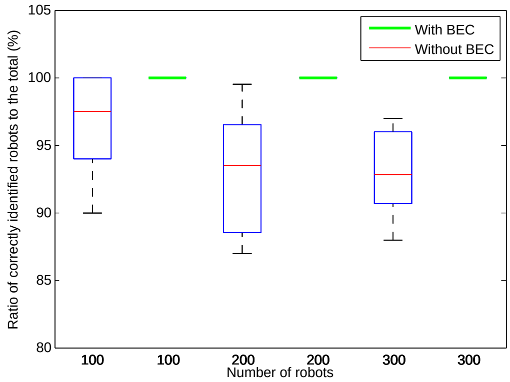


Figure 8. Accuracy of the online local boundary detection algorithm

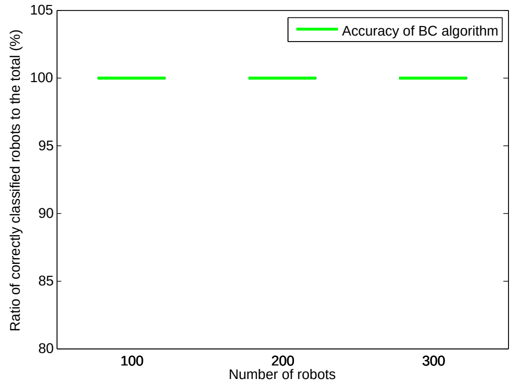
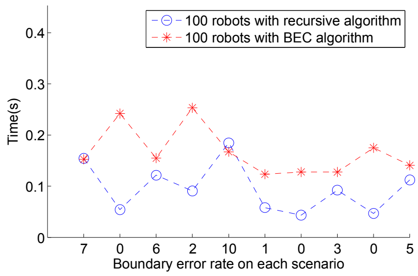
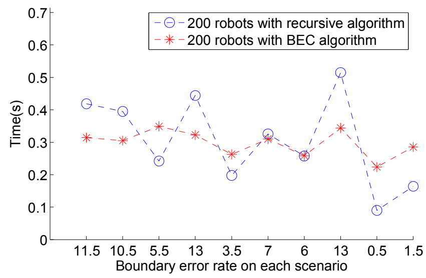


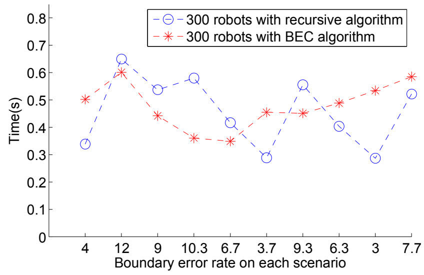
Figure 9. Accuracy of the boundary classification algorithm



(a) 100 nodes

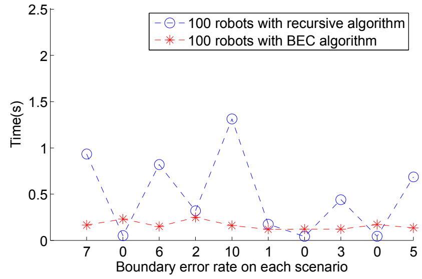


(b) 200 nodes

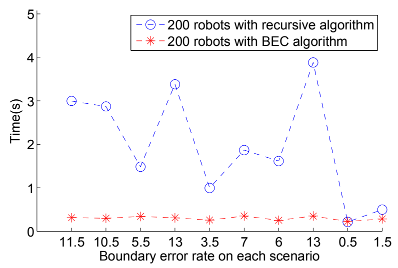


(c) 300 nodes

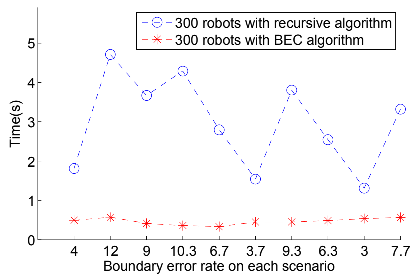
Figure 10. Time consuming for boundary detection with $T_m = 0.0125s$



(a) 100 nodes



(b) 200 nodes



(c) 300 nodes

Figure 11. Time consuming for boundary detection with $T_m = 0.125s$

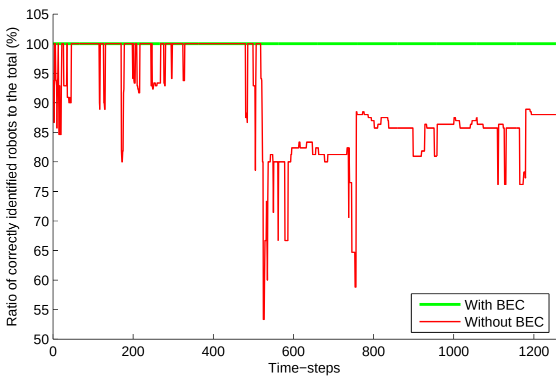


Figure 12. Accuracy of the boundary detection algorithm over time in simulation with 50 robots

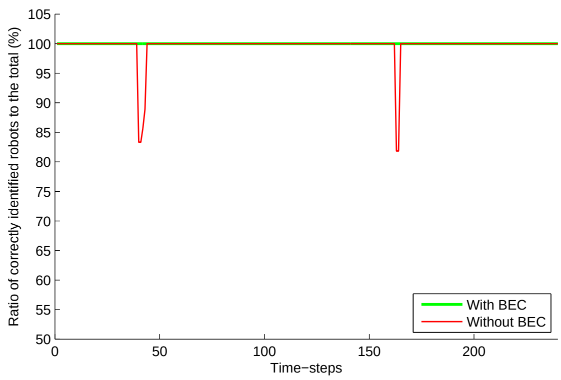


Figure 13. Accuracy of the boundary detection algorithm over time in real experiment with 14 robots

localisation within limited sensing range $r_c = 1m$. Two connected nearest neighbouring nodes can exchange information with delay communication time between them limited by T_m.

We have examined the effectiveness of the BEC and BC algorithms in both static and dynamic configuration established by an ad-hoc network of sensor nodes on a flat surface in 2D model.

Firstly, the BEC and BC algorithms are investigated in a static configuration with nodes that are spatially distributed with the Gaussian random distribution as described in [15] using 100, 200, and 300 nodes. Each algorithm was simulated in 10 different scenarios with delay communication time $T_m = 0.0125s$ and $T_m = 0.125s$ corresponding to the node speed rate (RSR) between node movement and communication $RSR = 0.001$ and $RSR = 0.01$ as examined in [11].

The simulation results show that accuracy of the local boundary detection algorithm with or without BEC algorithm as illustrated in Figure 8. The median accuracy of the algorithm without BEC is 97.50%, 93.50%, and 92.83% in cases 100, 200, and 300 nodes, respectively while the accuracy reaches 100% for all the cases when the BEC algorithm was applied. Figure 9 shows the accuracy of the BC algorithm. The result confirms that the algorithm successfully classifies boundaries for all the randomized scenarios.

The running time of the local boundary detection algorithm is shown in Figures 10 and 11. Because boundary error correction using recursive algorithm depends on number of boundary errors and requires complexity of communication $O(N)$, the running time with $T_m = 0.125s$ is greater than with $T_m = 0.0125s$ and it is more time consuming in scenarios with high boundary error rate. Unlike the recursive algorithm, the BEC algorithm has complexity of communication $O(1)$ so that it only depends on the status of neighbouring nodes. The results show that the BEC algorithm is faster than the recursive algorithm when $T_m = 0.0125s$ and the boundary error rate in the network is more than 6.6% while the running time of the BEC algorithm is much lower than the one of the recursive algorithm for all scenarios having the boundary error rate not equal zero with $T_m = 0.125s$.

Secondly, the BEC and BC algorithm were examined in a dynamic configuration with 50 mobile nodes. Boundaries were detected online while the nodes was deployed to track and occupy multiple targets as described in [15]. All nodes detect and classify themselves successfully overtime without any error. Accuracy of the boundary detection algorithm over time is shown in Figure 12. The video demonstration of the simulation can be seen at this link[1].

4.2 Experiments

In real-world experiments, we used 14 mobile robots as mobile sensor nodes to investigate the effectiveness of the online boundary detection and classification algorithm in a dynamic network. The robots are $14cm$ diameter disc-like differentially driven wheel platforms as in Figure 14. The sensing and communication range is set at $r_c = 1$ and the maximum velocity of the nodes is set at $0.8m/s$, similar to the simulation.

[1]https://youtu.be/RFf-jBtJL2k

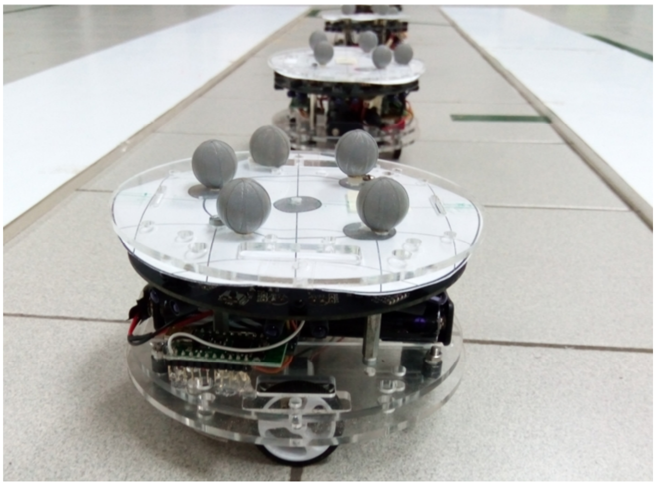


Figure 14. The real mobile robots used in the experiments

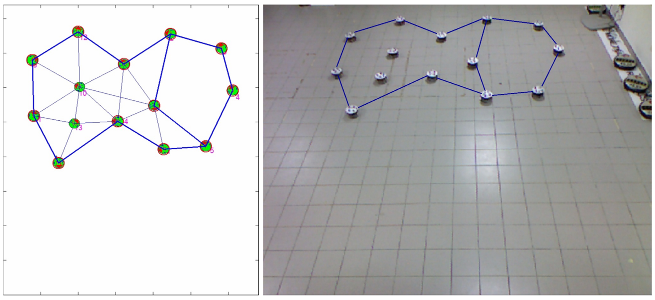


Figure 15. A boundary established in a network consisting of 14 robots: visualization (left) and the real system (right).

The online boundary detection and classification were measured while the robots was deployed on a flat surface to track and occupy multiple targets. We show that boundary nodes were successfully detected and classified with the accuracy 100% while the robots cooperatively tracked and occupied multiple targets as shown in Figure 13. The boundary was online created as illustrated in Figure 15 and the video demonstration can be seen at this link[2].

5 Conclusion

We have presented the online local boundary detection and classification algorithm with error correction based on the geometric approach in 2D model. Using this algorithm, all boundary errors in a network can be removed without recursive process and global synchronization. The accuracy of this new boundary detection and classification is 100%. We validated the quality of the new boundary detection and classification algorithm in simulations of both static and dynamic network configurations, and the real-world experiments with 14 mobile robots.

References

[1] S. K. Lee and J. McLurkin, “Distributed cohesive configuration control for swarm robots with boundary in-

[2]https://youtu.be/Vao9U8Z6Qxs

formation and network sensing," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 1161–1167.

[2] Y. Keung, B. Li, and Q. Zhang, "The intrusion detection in mobile sensor network," in *Proceedings of the eleventh ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2010, pp. 11–20.

[3] S. P. Fekete, A. Kröller, D. Pfisterer, S. Fischer, and C. Buschmann, *Neighborhood-Based Topology Recognition in Sensor Networks*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 123–136.

[4] A. Kröller, S. P. Fekete, D. Pfisterer, and S. Fischer, "Deterministic boundary recognition and topology extraction for large sensor networks," in *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*. Society for Industrial and Applied Mathematics, 2006, pp. 1000–1009.

[5] Y. Wang, J. Gao, and J. S. Mitchell, "Boundary recognition in sensor networks by topological methods," in *Proceedings of the 12th annual international conference on Mobile computing and networking*. ACm, 2006, pp. 122–133.

[6] O. Saukh, R. Sauter, M. Gauger, and P. J. Marrón, "On boundary recognition without location information in wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 6, no. 3, p. 20, 2010.

[7] W.-C. Chu and K.-F. Ssu, "Decentralized boundary detection without location information in wireless sensor networks," in *2012 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2012, pp. 1720–1724.

[8] A. Dabba and R. Beghdad, "Bcp: A border coverage protocol for wireless sensor networks," in *Science and Information Conference (SAI), 2014*. IEEE, 2014, pp. 632–640.

[9] R. Beghdad and A. Lamraoui, "Boundary and holes recognition in wireless sensor networks," *Journal of Innovation in Digital Ecosystems*, vol. 3, no. 1, pp. 1–14, 2016.

[10] L.-H. Zhao, W. Liu, H. Lei, R. Zhang, and Q. Tan, "The detection of boundary nodes and coverage holes in wireless sensor networks," *Mobile Information Systems*, vol. 2016, 2016.

[11] J. McLurkin and E. D. Demaine, "A distributed boundary detection algorithm for multi-robot systems," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2009, pp. 4791–4798.

[12] W.-C. Chu and K.-F. Ssu, "Location-free boundary detection in mobile wireless sensor networks with a distributed approach," *Computer Networks*, vol. 70, pp. 96 – 112, 2014.

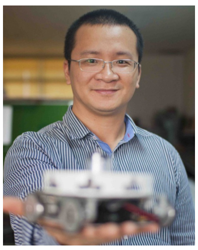
[13] B. Huang, W. Wu, G. Gao, and T. Zhang, "Recognizing boundaries in wireless sensor networks based on local connectivity information," *International Journal of Distributed Sensor Networks*, 2014.

[14] P. D. Hung, T. Q. Vinh, and T. D. Ngo, "An online local boundary detection and classification algorithm for networked multi-robot systems," in *2016 International Conference on Advanced Technologies for Communications (ATC)*, Oct 2016, pp. 253–258.

[15] ——, "A scalable, decentralised large-scale network of mobile robots for multi-target tracking," in *Intelligent Autonomous Systems 13*. Springer, 2014, pp. 621–637.

Pham Duy Hung received B.Sc. degree and M.Sc. degree in Electronics and Telecommunication from University of Engineering and Technology (UET), Hanoi, Vietnam in 2003 and 2006, respectively. He is currently pursuing the PhD degree in Robotics at UET. His research interests include distributed control of multi-agent systems with emphases of deployment, exploration, coverage, and multiple target tracking.

Tran Quang Vinh is an Associate Professor at the University of Engineering and Technology (UET), Hanoi. He is currently the head of department of Electronics and Computer Engineering. His interests include intelligent robotics, fuzzy logics, sensor fusion, networked robotics, computer architecture, instrumentation, and control engineering.



Trung Dung Ngo got Ph.D. degree in Electrical and Electronic Engineering (Robotics), Aalborg University in 2008, M.Sc. degree in Computer Systems Engineering (Robotics), University of Southern Denmark in 2004, and B.Sc. degree from Vietnam National University, Hanoi in 2000. He is currently with the University of Prince Edward Island (UPEI) where he is the founder and head of the More-Than-One Robotics laboratory (www.morelab.org) and the lead researcher of the Centre for Excellence in Robotics and Industrial Automation. Before joining UPEI, he was the faculty member of the Department of Electronic Systems, Aalborg University, and the Faculty of Science, University of Brunei. His research interests include multi-robot systems, modular robotics, and human-robot cooperation. He is an active member of IEEE.