Boundary Detection in a Swarm of Kilobots

Yingyi Kuang[1,2,⋆], Yuri Kaszubowski Lopes[1,3], and Roderich Groß[1]

[1] Sheffield Robotics, The University of Sheffield, Sheffield, UK
r.gross@sheffield.ac.uk
[2] Engineering and Applied Science School, Aston University, Birmingham, UK
180205312@aston.ac.uk
[3] Department of Software Engineering, Federal University of Technology – Paraná (UTFPR), Dois Vizinhos, Brazil yurilopes@utfpr.edu.br

Many tasks require swarms of robots to cover vast areas, for example, exploration, inspection, and search and rescue. A related problem is for the robots of the swarm to detect the boundary of their formation [5, 4]. Boundary detection can provide information about the extent to which an area is covered, including possible omissions, for example, caused by a sub-optimal distribution of robots or by inaccessible regions of the environment. Boundary detection is also relevant in other contexts, including sensor networks [3]. Topological approaches [1, 2, 8] estimate whether a node (e.g., a robot) is near the boundary without bearing or distance measurements. However, they often make strong assumptions about the distribution of nodes, and typically require computational and communication resources that exceed those found in swarms of miniature robots. To overcome these limitations, McLurkin and Demaine [6] proposed the cyclic-shape algorithm. Each robot determines whether its neighbours are interconnected. If they are—effectively forming a loop around the focal robot—the latter is considered as an interior node of the formation; otherwise, it is considered as a boundary node. One assumption of the cyclic-shape algorithm is that each robot can obtain distance and bearing measurements, and therefore has full information about the relative positions of nearby robots.

In this paper we present a local coordinate identification method that makes it possible for the cyclic-shape algorithm to be used on nodes that have distance but no bearing measurements. This makes it possible to perform distributed boundary detection on a physical swarm of Kilobots [7], which is a swarm robotics platform with exceedingly simple capabilities. Kilobots can exchange infra-red messages when nearby. They can use the signal strength to estimate the distance (but not the bearing) of other robots in their neighbourhood. In the following, we describe the identification method and report a series of systematic trials on a physical swarm demonstrating the feasibility of our solution.

The local coordinate identification method is based on the assumption that all robots in the swarm have a unique ID, can share a few bytes of information, and all have the same communication range. Each robot begins broadcasting its own ID. As the robot receives messages from its neighbours, it will store for every neighbour i, the corresponding ID and distance, d_i. In the second phase, the

⋆ The first author conducted the conceptual and experimental work while with The University of Sheffield. She is currently with Aston University.

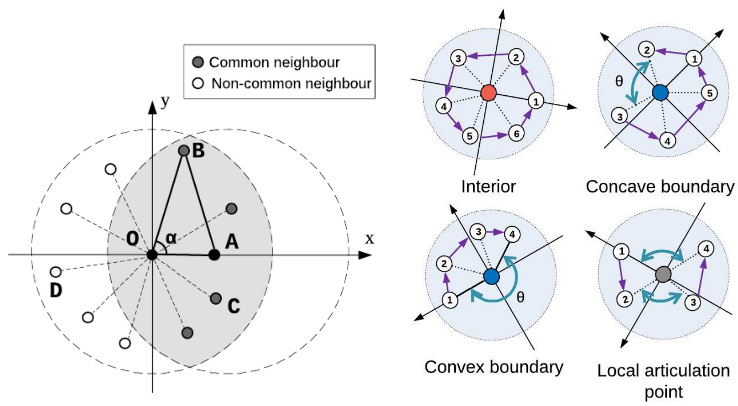
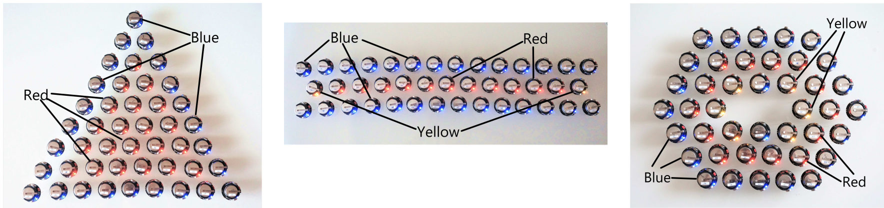


Fig. 1. Distributed boundary detection method. (a) The focal node (robot O) determines successively: the closest neighbour, A, defining the x-axis; furthest neighbour, B, that O and A have in common, defining the positive direction of the y-axis; other common neighbours, C; and non-common neighbours, D. (b) The focal node is of one of four possible types: it is interior (number of missing sectors $n = 0$), part of a concave boundary ($n = 1$ and $\theta < 180°$) or convex boundary ($n = 1$ and $\theta >= 180°$), or it is a local articulation point ($n > 1$), that is, a node connecting multiple networks.

robot (hereafter also called O) seeks to establish its own local coordinate system [see Figure 1(a)]. It identifies the closest neighbour, A, and places it at $(d_{OA}, 0)$, where d_{OA} is the distance between O and A. Then it identifies all neighbours it has in common with A, and chooses the one that is furthest to itself, neighbour B, as shown in Figure 1(a). The robot assumes that the triangle formed by O, A and B is not degenerate, and uses B to define the positive direction of the y-axis. In other words, B is located in either the first or second quadrant. To obtain B's coordinates, the robot calculates angle α, as shown in Figure 1(a), using trigonometry from d_{OA}, d_{OB} and d_{AB}, the latter being provided by A. The coordinates of B are then given as $(d_{OB} \cos \alpha, d_{OB} \sin \alpha)$. In the third phase, the robot calculates the coordinates of the remaining common neighbours. A common neighbour, C, forms a triangle with O and A. The first estimate of C's coordinates is obtained in the same way as for neighbour B. Robot C could be located on the same side of the x-axis as robot B (positive y-coordinate) or on the opposite side (negative y-coordinate). If B is outside of C's neighbourhood, or if the measured distance between B and C differs substantially from the predicted one, assuming they are located on the same side of the x-axis, the robot infers that B and C are located on opposite sides. In the final phase, the robot calculates the coordinates of the non-common neighbours. The robot chooses iteratively any non-common neighbour that forms a triangle with itself and any other neighbour for which the coordinates are already established, and then obtains the position information analogous to the aforementioned procedure.



(a) (b) (c)

Fig. 2. Kilobot robots densely arranged in one of three base shapes: (a) regular triangle, (b) narrow stripe, (c) hexagon with internal hole. Each robot displays the detection result using its on-board LEDs (red = interior, blue = convex boundary, yellow = concave boundary).

The robot assumes that the nodes are spaced sufficiently dense such that all positions can be estimated using this process.

Once the coordinates of its neighbourhood have been identified, the robot searches counter-clockwise through all its neighbours. It checks if each neighbour can communicate with the next one, that is, whether they are connected via an edge in the communication graph. The angles between pairs of subsequent connected neighbours are determined and summed up. If there exists no edge between subsequent neighbours, a *missing sector* has been detected. The cyclic-shape algorithm determines the number of missing sectors, n, and, if $n = 1$, the corresponding angle (θ). The procedure is illustrated in Figure 1(b).

To validate the boundary detection method on a real system, experiments with swarms of 41–45 Kilobot robots were conducted. The Kilobots were placed in an open environment, assuming one of three regular base shapes (see Figure 2). The robots were either placed in a "dense" (30 to 40 mm spacing) or "sparse" (45 to 55 mm spacing) setting; they were calibrated to have a communication range of 70 mm. Each of the six settings was tested 15 times. In other words, 90 trials were performed in total.

Figure 3 shows the classification accuracy for all six formations. The accuracy is defined as the percentage of robots that identify their boundary type correctly. It is between 83% and 89%, depending on the formation. Irregular shapes were tested too, and led to a decreased accuracy, in between 20% and 70%. Further analysis showed that the interior nodes had a lower classification accuracy compared to the boundary nodes (around 15% less on average). We further investigated the coordinates, obtained via a serial port from robots with incorrectly classified type. Errors were mainly caused by inaccuracies of distance measures and the non-detection of neighbours. Future work will address these uncertainties, for example, by using filtering algorithms or allowing robots to exchange information about their respective boundary types. This could make the system more robust to variations in shape, sensing noise, and dropped messages.

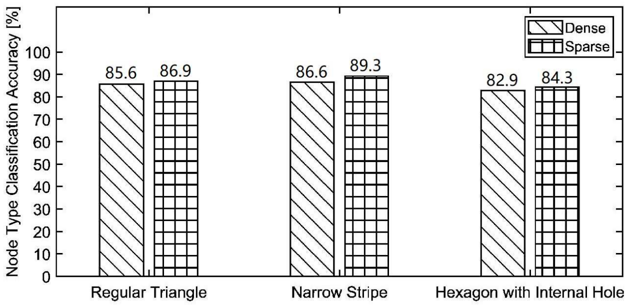


Fig. 3. Experimental results. Node type classification accuracy (mean over 15 trials per box) for different shapes with the robots either densely or sparsely distributed.

Acknowledgements

This research was supported by the Engineering and Physical Sciences Research Council (grant No. EP/J013714/1).

References

1. Fekete, S.P., Kröller, A., Pfisterer, D., Fischer, S., Buschmann, C.: Neighborhood-based topology recognition in sensor networks. In: International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics. pp. 123–136. Springer (2004)
2. Funke, S.: Topological hole detection in wireless sensor networks and its applications. In: Proceedings of the 2005 Joint Workshop on Foundations of Mobile Computing. pp. 44–53. ACM (2005)
3. Khan, I., Mokhtar, H., Merabti, M.: A survey of boundary detection algorithms for sensor networks. In: Proceedings of the 9th Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting (2008)
4. Lee, S.K., Kim, D., Shin, D.S., Jang, T., McLurkin, J.: Distributed deformable configuration control for multi-robot systems. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 5347–5354. IEEE (2016)
5. Lee, S.K., McLurkin, J.: Distributed cohesive configuration control for swarm robots with boundary information and network sensing. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 1161–1167. IEEE (2014)
6. McLurkin, J., Demaine, E.D.: A distributed boundary detection algorithm for multi-robot systems. In: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4791–4798. IEEE (2009)
7. Rubenstein, M., Ahler, C., Nagpal, R.: Kilobot: A low cost scalable robot system for collective behaviors. In: 2012 IEEE International Conference on Robotics and Automation (ICRA). pp. 3293–3298. IEEE (2012)
8. Wang, Y., Gao, J., Mitchell, J.S.: Boundary recognition in sensor networks by topological methods. In: Proceedings of the 12th Annual International Conference on Mobile Computing and Networking. pp. 122–133. ACM (2006)