Approximate Characterization of Multi-Robot Swarm “Shapes” in Sublinear-Time

Lantao Liu, Benjamin Fine, Dylan Shell and Andreas Klappenecker
Dept. of Computer Science and Engineering
Texas A&M University, College Station, TX, USA
{lantao, fineb, dshell, klappi}@cse.tamu.edu

Abstract—Many envisioned applications of multi-robot swarms involve the detection, production or maintenance of global structures through only local means. This paper introduces a scalable, distributed algorithm to approximately characterize important global geometric and topological properties. For a given spatial arrangement of robots, the algorithm estimates the longest network (geodesic) distance in any direction as well as the average Euclidean distance only using locally sensed information. In so doing, the robots need only to communicate with and sense (range and bearing) nearby robots. The algorithm uses a greedy method to approximate both distance metrics via parallel one-way message traversals. We provide a bound for the number of such traversals, showing a global characterization is produced in a running time that is sublinear in the total number of robots. Along with this analysis, we conduct simulations with hundreds of robots to validate the algorithm.

I. INTRODUCTION

In recent years, multi-robot systems (MRS) with increasingly large numbers of robots have been demonstrated in research laboratories. Hardware for MRS with thousands of robots will soon become a reality. Oftentimes robots are required to collectively determine certain global, task-related properties. This must be done efficiently despite each robot having to sense, act and plan independently. We believe, with increasing system sizes, even algorithms with running times linear in the number of robots, will be too slow for dynamic applications.

The goal of this paper is to show that useful global geometric and topological properties, which sketch the “shapes” of a whole system, can be approximated using only a subset of robots and limited communications. More specifically, we are interested in measuring the geodesic distance which reflects the minimum number of communication hops among two robots in a distributed network and the path distance, which is the sum of Euclidean edge lengths along a traversed path. The geodesic distance provides an estimate of the message broadcast time and the path distance measures the distance of the two most distant robots along a given direction. Therefore, these two metrics are useful for describing the current system’s formation.

II. RELATED WORK

In order for future algorithms to scale to large-scale MRS, the algorithms should establish the link between individual robots and their global properties without involving every robot. In the last decade there has been growing interest in sublinear time algorithms [1], [2], [3] which were originally designed to solve massive data processing problems. Many property querying or testing algorithms [4], [5], [6] use sublinearity in order to test or query properties of massive graphs. To the best of our knowledge, there has not been any work done on sublinear time algorithms applied in distributed MRS, with only some work in randomized algorithms for robotic motion and path planning. It is worth emphasizing, there are several challenges which arise in attempting to apply existing property testing algorithms to MRS.

In MRS, robots only have the ability to communicate with neighboring robots within some limited range, therefore the information a robot can gain about the global state of the MRS is limited. This makes for a fundamentally different data access model than generally used for property testing. Most of the work done on property estimation is in a centralized context. Works such as [1], [7], [8], use either an adjacency matrix or an incidence list to represent dense and sparse graphs, respectively. Randomly sampling from these structures is very different from an individual robot sampling its local communication graph. Furthermore, these works do not take advantage of the parallelism offered by MRS. For the purposes of this work we are interested in the situations where the size of the system is extremely large (i.e., many hundreds of robots or even more) and the environment is unknown prior to the execution of the system. In such cases, it is not possible to set up the required type of communication for the above global data structures.

The present work introduces an example of an algorithm concerned with both network and embedding properties that employs a message passing strategy which is easily implemented on physical robots and is able to exploit parallel phases of computation and communication.

III. PRELIMINARIES

We aim to develop algorithms for MRS with many hundreds of robots. Although such systems do not yet exist, we believe considering currently deployed systems is necessary in order to build a theoretical model of the capabilities one may reasonably expect such future robot systems to have. For this purpose, we have considered the e-puck [9] as it is a good representative class of swarm robot systems: indeed at 70mm in size, 200g in weight, and relatively low cost, it is an ideal robot for large-scale MRS. Most important for this paper is the array of 8 infra-red proximity sensors,
which can be used for communication [10]. A compass and the Range & Bearing Miniaturized Board [11] permits the robot to identify, detect and communicate with a small subset of neighboring robots. Along with communication the device permits the robot to estimate a relative range and bearing to each neighbor. As with all robotics, sensing and communication involves noise and failure. Note that our simulations explicitly consider the effect of noise by perturbing input values provided to the robots.

Due to the spatial and sensing/communication restrictions, we model the MRS as a sparse graph, where the vertices represent the robots and the edges represent the sensing/communication connections. We assume each robot has bidirectional communication, therefore we can analyze the system as an undirected connected graph $G = (V, E)$, where $|V| = n$ and $e(u, v) \in E$ is an edge between vertex $u \in V$ and $v \in V$. The algorithm does not require multi-hop or addressed communication. We do assume, any message that is delivered, does not undergo corruption en route. These communication assumptions are easily relaxed since the bounds below are written in terms of expected packets delivered.

Definition 3.1: The geodesic distance between two vertices $u$ and $v$ is denoted by $d_G(u, v)$ and is calculated as the total number of communication edges in the geodesic shortest path between $u$ and $v$.

Definition 3.2: The weight of an edge in the graph is the sensed distance between $u$ and $v$. The weighted path distance (path distance) between $u$ and $v$ is denoted by $d_P(u, v)$ and is the sum of all the edge weights in the shortest path from $u$ to $v$.

Definition 3.3: The geodesic diameter ($D_G$) of a graph $G$ is defined as $\max_{u,v\in V} d_G(u, v)$. Similarly, we define $D_P = \max_{u,v\in V} d_P(u, v)$ to refer to the path diameter which can approximate the physical size of the topology.

Definition 3.4: We use $\text{deg}(v)$ to denote the degree of vertex $v$, which means the number of directly connected neighboring vertices with $v$.

Definition 3.5: An R-disk is a disk of radius $R$ centered on a given robot. We assume the R-disk graph on $G$ is connected; otherwise, we can consider each connected component separately.

Definition 3.6: For any real number $\epsilon > 0$, a $(1 + \epsilon)$-approximation of a quantity $Q : G \rightarrow (0, \infty)$ is an algorithm, which on input $G$, with probability at least $2/3$ outputs a value in the interval $[Q, (1 + \epsilon)Q]$.

We use the notational convenience of dividing the system into two disjoint sets $S_p$ and $S_i$, where $S_p$ is formed from all vertices on the periphery (i.e., the “hull”) of the graph and $S_i$ contains all the remaining inner vertices.

IV. CHARACTERIZING SWARM SHAPES

A. Algorithm Stages

Our distance approximation approach is straightforward and the principle idea is easiest to understand from the continuum limit perspective. Imagine the large-scale MRS is encircled by a differentiable periphery curve. One can then uniformly sample $q$ points on the periphery and calculate the inward pointing normal as orthogonal to the tangent for each point. The normals here are the global directions for messages to traverse the system. Since the periphery is continuously differentiable and closed, the normal directions are expected to cover from 0 to $2\pi$, therefore, likely to capture the traversals in every direction, as illustrated in Fig. 1. A high-quality approximation of the shape requires only that $q$ constitute a sufficient number of samples.

However, in practice the system is modelled to a graph without a differentiable perimeter. We approximate the “normal” direction by averaging the directions of its incoming and outgoing boundary edges. Actual measurements of the MRS are achieved by message-passing in the communication network, with two kinds of message:

Normal Traversal Message (NTM) contains a determined normal traversal direction, a counter of visited robots (to calculate $d_G$), an odometer (to record $d_P$), an angular accumulator (for calculating the angular deviations) and a history of visited robots.

Peripheral Traversal Message (PTM) contains a peripheral robots counter, the probability to trigger the NTM and a short queued history of recently visited peripheral robots.

Stage 1: We denote by $a_1$ the robot from the set $S_p$ from which the shape query is initiated, e.g., arbitrarily selected by an outside (human) operator. Robot $a_1$ then initiates and sends a PTM to the nearest peripheral robot $a_2$, which serves to determine the passing direction; assume counter-clockwise for the remainder of this work. To search the successive peripheral neighbor, each robot chooses the edge which is the right or left most among all its outgoing edges, depending on the passing direction. The PTM continues until it returns to $a_1$ after each robot along the loop is flagged as a peripheral robot in $S_p$ and each records its predecessor and successor along the “chain”. Finally the size of $S_p$, denoted $\tilde{n}$, is obtained.

Stage 2: After robot $a_1$ obtains $\tilde{n}$, the sampling probability $P_s$ is determined via dividing sample size $q$ (discussed in Section IV-C) by $\tilde{n}$ and saved in the PTM. The message is then resent along the periphery. Each periphery robot which receives a PTM with $P_s$ immediately passes it to its successor and runs a random generator to decide with probability $P_s$ if it should initiate a NTM. If a NTM is to be sent, the robot initiates it with $d_G = 0$, $d_P = 0$ and sends it to a neighbor in the normal direction. A NTM ends its life cycle of normal traversal when it reaches a new periphery robot. Each robot in $S_i$ that receives the NTM updates the odometer records of $d_G$ and $d_P$ as well as the information of angular deviation and traversed history.

Stage 3: Any robot in $S_p$ immediately passes any received message to its peripheral successor along the determined peripheral “chain.” Robot $a_1$ collects all of the packages and ultimately computes approximate shapes from the two distance metrics.

Stage 4: The results are broadcast via any appropriate propagation strategies to the robots requiring the approx-
direction is the polar axis $L$ coordinate system to describe the positions of the robots. In ing/communication radius $R$ however each edge length is bounded by a constant sensitivity.

There is a one-way transversal path which can approximate to the normal transversal direction. We want to investigate if we can determine its successor neighbor by selecting the robot nearest from $\theta_i \in [-\gamma/2, \gamma/2]$. Here, $\gamma$ is the angle within which the message can be passed. Hence, the message can be passed only to the robots inside the $[-\gamma/2, \gamma/2]$ region, which we term as the $R-\gamma$-sector, see Figure 2(b).

**R-\gamma-sector algorithm:** Inside the $R-\gamma$-sector, the robot at pole $O$ always chooses the neighbor which is farthest from $O$ to be the NTM successor. If there are no robots in the $R-\gamma$-sector, then the neighbor closest to the normal will be chosen.

We are interested in the constraints of the angle $\gamma$ as well as the radius $R$ with regard to some specific $(1 + \epsilon)$-approximation.

**Theorem 4.1:** Given a large set of robots which are uniformly distributed in Euclidean space, the path of $R-\gamma$-sector algorithm approximates the geodesic distance and shortest path distance well, given there is at least one robot in the $R-\gamma$-sector.

**Proof:** It is difficult to prove this fact using deterministic methods, therefore we will analyze the algorithm from the stochastic perspective. More specifically, we investigate the conditions which can bound the algorithm’s approximation of the optimal geodesic distance and shortest path distance.

Since in any local area the robots are uniformly distributed, thus the $r_i$ (as well as $\theta_i$) are i.i.d random variables. Between any two robots with a geodesic distance greater than 1, the lower bound of the geodesic distance $d_{GB}$ in a specific direction can be expressed as $d_{GB} = d_{E0}/R$, where $d_{E0}$ is the Euclidean distance between the two ending robots. Suppose the $R-\gamma$-sector algorithm returns $\hat{d}'_G$, then we seek a bound in the form

$$d'_G \leq (1 + \epsilon) \cdot d_{GB},$$

where $d'_G$ is the number of edges in the path, calculated using the NTM Euclidean distance, $d_{E0}$, divided by the expectation of each edge projection on $d_{E0}$:

$$d'_G = \frac{d_{E0}}{E[r_{i_{max}} \cdot \cos(\theta_i)]} = \frac{d_{E0}}{E[r_{i_{max}} \cdot E[\cos(\theta_i)]},$$

where

$$E[\cos(\theta_i)] = \int_{-\gamma/2}^{\gamma/2} \cos(\theta_i) \cdot \frac{1}{\gamma} \cdot d\theta_i = \frac{2}{\gamma} \cdot \sin\frac{\gamma}{2}. \quad (3)$$

The expectation of the farthest distance in a $R-\gamma$-sector (as the selection of $a$ in Fig. 2(b)), denoted by $E[r_{i_{max}}]$, can be calculated by using order statistics. Assume $N$ is the number of robots in the $[-\gamma/2, \gamma/2]$ $R-\gamma$-sector. The event $r_{i_{max}} \leq t$ is equivalent to the event that $r_i \leq t, \forall i \leq N$. Therefore $r_{i_{max}}$ has distribution function

$$F_{max}(t) = t^N, \quad (4)$$

where $F(t) = t/R$ is the distribution for each $r_i, i \in [1, N]$. The expectation of $r_{i_{max}}$ can be obtained by

$$E[r_{i_{max}}] = \int_0^R t \cdot dF_{max}(t) = \frac{N \cdot R}{N + 1} \quad (5)$$

Combining (1)—(5), we get the relation

$$\frac{N}{N + 1} \cdot \frac{\sin(\gamma/2)}{(\gamma/2)} \geq \frac{1}{1 + \epsilon}. \quad (6)$$

Next we want to compare the path distance between the $R-\gamma$-sector algorithm and the lower bound $d_{E0}$. The path distance $d'_P$ from our algorithm is

$$d'_P = E[\sum_{i=1}^{\Delta'_P} r_{i_{max}}] = \sum_{i=1}^{\Delta'_P} E[r_{i_{max}}] = \Delta'_P \cdot E[r_{i_{max}}],$$

where $\Delta'_P$ is the number of edges, $\Delta'_P = \hat{d}'_G$, thus

$$d'_P = \frac{\hat{d}'_G}{E[\cos(\theta_i)]}. \quad (8)$$

To bound $d'_P$ within a $(1 + \epsilon)$-approximation, let $d'_P \leq (1 + \epsilon) \cdot d_{E0}$, therefore

$$\frac{\sin(\gamma/2)}{(\gamma/2)} \geq \frac{1}{1 + \epsilon}. \quad (9)$$

Equations (6) and (9) are two bounds for $(1 + \epsilon)$-approximations of geodesic distance and shortest path distance, respectively. The comparison between them reveals that (6) is the more constrained and bounds both approximations. Moreover, if $N$ is large enough, $(N/(N + 1) \rightarrow 1$, then (6) and (9) are approximately equal, which suggests our algorithm can simultaneously $(1 + \epsilon)$-approximate the geodesic distance and the shortest path distance.
C. Bounding the Number of NTM Samples

In this section we investigate the lower bound of NTM samples required in $S_p$ to capture the distribution of the distance metrics. We have shown in the previous sections, our approach can approximate the geodesic distance and shortest path distance following a traversal direction under certain densities. Here we use the geodesic distance metric to analyze the minimal number of samples required. The main idea is from the perspective of comparing the averaged distances obtained from the chosen samples, with the theoretical expectation of all robots in $S_p$. In other words, we want to bound the probability that the average of our outputs satisfies the $(1 + \epsilon)$-approximation.

We are inspired by the work of [1]. The difference between the MRS and a common graph problem is the communication graph is completely unknown before the message traversals and this needs the robots to exploit the graph using local sensing information along the traversals. Also the robots are distributed, which means approximation tasks can be executed in parallel. Here we assume robots on the periphery are also distributed uniformly, which guarantees well distributed paths if we uniformly sample a subset of robots among them.

Theorem 4.2: A $(1 + \epsilon)$-approximation of the average distance for a distributed MRS can be achieved by using a sample size $q = \Theta(\sqrt{2}e^{-2}2^{n})$, where $n = |S_p|$.

Proof: Assume the geodesic distance $d$ between any two peripheral robots along the normal traversal direction satisfies $d \in [\bar{d}, \hat{d}]$, where $\bar{d}$ and $\hat{d}$ are the shortest and longest geodesic distances between any two peripheral robots, respectively. Let $\Delta d = \hat{d} - \bar{d}$, and $p_i$ denote the fraction of paths with geodesic distance $d_i = \hat{d} + i$ in all $\hat{n}$ paths $(i \in [0, \Delta d])$. Then let $\eta$ denote a random variable which takes value $d_i$ with probability $p_i$ and let $\eta_1, \eta_2, \ldots, \eta_q$ be independent random variables which are distributed the same as $\eta$. From the definition of $(1 + \epsilon)$-approximation, we bound the probability that the average of the output, $\eta = \sum_{j=1}^{q} \eta_j$, deviates from its expectation $E[\eta]$ by more than $\epsilon E[\eta]$. From Chebyshev’s inequality, we have:

$$\Pr \left[ q^{-1} \sum_{j=1}^{q} \eta_j - E[\eta] \geq \epsilon E[\eta] \right] \leq \frac{E[\eta^2] - E[\eta]^2}{q^2 E[\eta]^2}$$  \hspace{1cm} (10)

Since $\eta$ is integrally ranged in $[\bar{d}, \hat{d}]$, following the definition of $E[\eta^2]$ yields

$$E[\eta^2] = \sum_{i=0}^{\Delta d} p_i \cdot (\hat{d} + i)^2 \leq \hat{d} \cdot E[\eta].$$  \hspace{1cm} (11)

Here $p_i$ represents the fraction of a subset of vertices in all peripheral vertices, so we have $p_i \geq 1/n$, and consequently

$$E[\eta] = \sum_{i=0}^{\Delta d} p_i (\hat{d} + i) \geq \frac{1}{n} \frac{(\hat{d} + \bar{d}) \Delta d}{2} = \frac{\hat{d}^2 - \bar{d}^2}{2n}. \hspace{1cm} (12)$$

Since we assume $E[\eta] \geq \hat{d} \geq 1$, namely $-\bar{d}^2 \geq -E[\eta]^2$, we may substitute into (12) obtaining

$$2\hat{n} \cdot E[\eta] + E[\eta]^2 \geq \hat{d}^2. \hspace{1cm} (13)$$

Combining (11) and (13) to cancel $\hat{d}$ we get

$$E[\eta^2] \leq 2\hat{n} \cdot E[\eta]^3 + E[\eta]^4 \leq (2\hat{n} + 1) \cdot E[\eta]^4. \hspace{1cm} (14)$$

Note the relaxation of (14) is because $E[\eta] \geq 1$. Therefore,

$$E[\eta^2] \leq \sqrt{2\hat{n} + 1} \cdot E[\eta]^2. \hspace{1cm} (15)$$

From Definition 3.6, to obtain a $(1 + \epsilon)$-approximation of $E[\eta]$, means

$$\Pr \left[ q^{-1} \sum_{j=1}^{q} \eta_j - E[\eta] \geq \epsilon \cdot E[\eta] \right] \leq \frac{1}{3}. \hspace{1cm} (16)$$

Combining (15) and (10) we obtain

$$q \geq 3e^2(\sqrt{2\hat{n} + 1} - 1). \hspace{1cm} (17)$$

Therefore, a sample of size $q = \Theta(3\sqrt{2}\epsilon^{-2}\sqrt{n})$ will suffice to approximate the average distance.

D. Running Time Analysis

The overall running time is $O(\hat{n})$, which is sublinear with regard to $O(n)$. The time complexity is measured by the time steps of message passing hops. In Stages 1 and 2, there are two rounds of sequential message passing in the set of $S_p$, so the running time is $O(\hat{n})$. Also in Stage 2, because the path along the normal direction in $S_i$ is shorter than the length of the perimeter of the network, and the message passing implemented in parallel, the running time in this part of Stage 2 is also $O(\hat{n})$. In Stage 3, there is at most one more round of sequential message passing in $S_p$, so the running time is still $O(\hat{n})$, even considering the overhead in Stage 2. The propagation procedure in Stage 4 can also be controlled within $O(\hat{n})$ by broadcasting from $S_p$ to $S_i$ in parallel. Therefore, the overall running time is $O(\hat{n})$, which is determined by the number of peripheral robots for a given uniformly distributed MRS.

V. Experiments

We simulated the algorithm with hundreds of robots in order to validate the approach. To permit easy visualization of the message passing results, we generated the networked topology in a 2-dimensional plane. All the robots are homogeneous and have identical sensing and communication ranges. Within the sensing/communication range, each robot is capable of recognizing its neighbors as well as their distances and bearings. In order to simulate the errors, we perturb each neighbor’s sensed distance by a random value in the range $[-20cm, 20cm]$ and perturb every neighbor’s sensed bearing by a random value in the range $[-5^\circ, 5^\circ]$. All information is collected and gathered via the PTM and NTM message packages. The multi-robot topology is generated by randomly perturbing well arranged formations; topologies are controlled to give different typical shapes such as circles, ellipses, dumbbells and squares; see Figure 3. The simulation environment is a 100m$\times$100m square. The density, as well as the degree, is controlled by adjusting the total number of robots in specific sized areas. To simplify the analysis, we set the diameter of all shapes to be 100 meters, which means the major axis of a ellipse, the longest axis of the dumbbell and the diagonals of a square are all equal to 100 meters. The results below are the mean values of 10 sets of experimental data.
Table I lists the diameter testing results of the three typical shapes. The measurements include the geodesic diameter $D_G$, the $\epsilon$ value $\epsilon_D G$ with regard to lower bound $D_G$, and the standard deviation of angles $\sigma_{DG}$, indicating deviation from the true diameter directions. Similarly, $D_P$, $\epsilon_D P$ and $\sigma_{DP}$ are the counterparts of shortest path distances. We first fixed the $\gamma$ angle to be 30° and observed the performance while varying the degree of the graph. Then we fixed the degree at 20 and observed the performance as $\gamma$ varies. In Table I, the diameter variances and angular deviations indicate the algorithm estimates both the geodesic and shortest path diameters relatively well. The results also imply that the larger the degree, the better the path quality; see Figure 4. However, the increasing $\gamma$ causes deterioration in the measurements, especially in the shortest path distance. In addition, the results of the ellipse and dumbbell are better than the square, implying the algorithm works better for shapes with smooth peripheries. Notice that the diameter is just a special distance among all distances of every direction, which were also measured and can be queried via the algorithm.

Table II is the comparison of NTM passing between the $R_{\gamma}$-sector method ($R_{\gamma}$) and normal-closest the (NC) method which greedily chooses the robots which are closest to the normal direction. We fix the $\gamma$ angle to be 30° and gradually increase the degrees, then compare the $D_G$ and $D_P$. The results indicate the $R_{\gamma}$ method approximates the NC method well and becomes increasingly accurate as density increases. Figure 5 also shows the similarity of the shortest paths from the two methods. The $R_{\gamma}$ method is better than the NC method in that, as the density increases, the geodesic diameter $D_G$ becomes much more accurate (actually converges to the bound) than NC method (diverges from the bound).

Table III shows the effect of changing the number of NTM samples. This is directly determined by the sampling probability $P_s$, which is obtained by setting the $\epsilon$ value. The results show, as $\epsilon$ increases, both the geodesic diameter and path diameter decrease, and some geodesic diameters even fall below the lower bound as the approximation becomes coarse and inaccurate. In addition, as $\epsilon$ increases, the average distances $d_G$ and $d_P$ oscillate with increasing amplitude and tend to diverge, which is indicated by the standard deviations $\sigma_{DG}$ and $\sigma_{DP}$, respectively. In Table III, the reason $P_s = 1.0$ when $\epsilon \leq 0.6$ is that the number of peripheral robots is insufficient (currently our simulation supports ~1,000 distributed robots). The bound derived in Section IV-C is particularly useful when $\tilde{n}$ is very large.

Table IV shows the number of peripheral robots and total time steps used in a MRS with 1,000 robots. Each time step represents the time for a sequential communication hop or
TABLE III
BOUNDING NUMBER OF NTM SAMPLES

<table>
<thead>
<tr>
<th>Shape</th>
<th>Measure</th>
<th>ϵ=0.3</th>
<th>ϵ=0.6</th>
<th>ϵ=0.9</th>
<th>ϵ=1.2</th>
<th>ϵ=1.5</th>
<th>ϵ=1.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ellipse</td>
<td>D_p</td>
<td>1.0</td>
<td>1.0</td>
<td>0.75</td>
<td>0.45</td>
<td>0.25</td>
<td>0.17</td>
</tr>
<tr>
<td></td>
<td>D_Q</td>
<td>15.2</td>
<td>15.2</td>
<td>14.1</td>
<td>14.0</td>
<td>13.6</td>
<td>13.6</td>
</tr>
<tr>
<td></td>
<td>D_P</td>
<td>101</td>
<td>101</td>
<td>98.6</td>
<td>98.2</td>
<td>98.5</td>
<td>98.5</td>
</tr>
<tr>
<td></td>
<td>d_G (\sigma_d G)</td>
<td>10.3 (0.2)</td>
<td>10.3 (0.2)</td>
<td>10.3 (0.4)</td>
<td>10.0 (0.5)</td>
<td>10.5 (0.3)</td>
<td>10.7 (1.1)</td>
</tr>
<tr>
<td></td>
<td>d_P (\sigma_d P)</td>
<td>68.7 (2.9)</td>
<td>68.7 (2.9)</td>
<td>67.6 (5.1)</td>
<td>67.7 (7.7)</td>
<td>70.4 (8.0)</td>
<td>70.0 (11.4)</td>
</tr>
<tr>
<td>Dumbbell</td>
<td>D_p</td>
<td>1.0</td>
<td>1.0</td>
<td>0.75</td>
<td>0.45</td>
<td>0.25</td>
<td>0.18</td>
</tr>
<tr>
<td></td>
<td>D_Q</td>
<td>15.9</td>
<td>15.9</td>
<td>15.2</td>
<td>14.0</td>
<td>12.3</td>
<td>12.3</td>
</tr>
<tr>
<td></td>
<td>D_P</td>
<td>102</td>
<td>102</td>
<td>100</td>
<td>99.7</td>
<td>98.3</td>
<td>96.9</td>
</tr>
<tr>
<td></td>
<td>d_G (\sigma_d G)</td>
<td>9.7 (0.5)</td>
<td>9.7 (0.5)</td>
<td>10.4 (0.6)</td>
<td>9.5 (0.7)</td>
<td>9.4 (1.2)</td>
<td>8.9 (1.7)</td>
</tr>
<tr>
<td></td>
<td>d_P (\sigma_d P)</td>
<td>64.3 (3.5)</td>
<td>64.3 (3.5)</td>
<td>63.6 (6.2)</td>
<td>62.1 (8.2)</td>
<td>67.8 (7.4)</td>
<td>59.5 (9.3)</td>
</tr>
<tr>
<td>Square</td>
<td>D_p</td>
<td>1.0</td>
<td>1.0</td>
<td>0.68</td>
<td>0.38</td>
<td>0.25</td>
<td>0.17</td>
</tr>
<tr>
<td></td>
<td>D_Q</td>
<td>14.4</td>
<td>14.4</td>
<td>13.6</td>
<td>12.9</td>
<td>12.2</td>
<td>11.7</td>
</tr>
<tr>
<td></td>
<td>D_P</td>
<td>98.9</td>
<td>98.9</td>
<td>92.1</td>
<td>88.4</td>
<td>78.0</td>
<td>80.4</td>
</tr>
<tr>
<td></td>
<td>d_G (\sigma_d G)</td>
<td>10.2 (0.3)</td>
<td>10.2 (0.3)</td>
<td>10.0 (0.9)</td>
<td>10.7 (2.1)</td>
<td>10.3 (1.7)</td>
<td>9.2 (2.8)</td>
</tr>
<tr>
<td></td>
<td>d_P (\sigma_d P)</td>
<td>70.4 (4.3)</td>
<td>70.4 (4.1)</td>
<td>71.2 (7.5)</td>
<td>71.8 (7.7)</td>
<td>72.0 (8.8)</td>
<td>66.0 (15.2)</td>
</tr>
</tbody>
</table>

In each experiment we use 1,000 robots. The sensing range \( R = 8m \).

TABLE IV
TIME COMPLEXITY TESTING WITH 1000 ROBOTS

<table>
<thead>
<tr>
<th>Radius</th>
<th>Measurement</th>
<th>Circle</th>
<th>Ellipse</th>
<th>Dumbbell</th>
<th>Square</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R \approx 8m )</td>
<td>[(</td>
<td>S</td>
<td>) ( t \times ) steps]</td>
<td>65</td>
<td>54</td>
</tr>
<tr>
<td>(deg%0)</td>
<td></td>
<td>187</td>
<td>157</td>
<td>162</td>
<td>170</td>
</tr>
<tr>
<td>( R \approx 6m )</td>
<td>[(</td>
<td>S</td>
<td>) ( t \times ) steps]</td>
<td>87</td>
<td>70</td>
</tr>
<tr>
<td>(deg%15)</td>
<td></td>
<td>246</td>
<td>203</td>
<td>214</td>
<td>226</td>
</tr>
</tbody>
</table>

multiple parallel hops (a hop includes the time of processing the message data and passing it to the successor), and thus reflects the time complexity. The total time steps shown in Table IV records the first 3 stages described in Section IV-A. We show results with two different average degrees, obtained by adjusting the sensing/communication radius \( R \) while fixing the robots size and density. The results indicate that shorter sensing/communication distance increases the time complexity. However, as long as the network is connected, the total time steps are much less than the total number of robots.

VI. DISCUSSION

The focus of our work is on using only simple local communication and sensing to provide approximate characterizations of global geometric shape without involving every robot. It is worth pointing out, an alternative approach would be to employ robot to robot pairwise measurements around the periphery and simply connecting these positions after the first phase of the algorithm. Doing so would accumulate errors and any small errors at first could be gradually magnified for large-scale MRS, which would greatly distort the shape. This requires the method to have either reliable localization for each robot or some relaxation-based method to redistribute the error. Should a coarse global shape descriptor be adequate, we believe the proposed algorithm, based on the independent sensing and communication, to be superior. Moreover, communication at different ranges may permit an extension of the present work to yield estimates without involving every robot on the periphery.

It should be noted, the proposed algorithm can approximate the average distances (both geodesic and path distance) well but it may not successfully capture the diameters if the shape is very irregular, e.g., a hook-like shape. When there is no NTM path along the real diameter direction, this will not be captured by the algorithm. Analysis of the distribution and/or local structure in this distribution could however, permit one to construct a signature for such irregular shapes.

To improve accuracy of the approximation and further reduce the time complexity, there are several worthy directions for future work. One may reconstruct the shape of the swarm by first computing the convex-hull, by using information from NTMs and the inaccurate recursive positions mentioned above. Additionally, one may improve the sampling strategy by considering the peripheral curve’s “derivative” (more samples in the sharp corners and less in the parts with limited curvature). Certainly insights will likely be gained about real systems by considering explicitly models of sensor noise (or communication unreliability) and their effect on the bound of accuracy or the numbers of samples required.

VII. CONCLUSION

This paper introduces a simple algorithm which uses limited sensing information and a sublinear amount of time to approximately characterize the shapes of a multi-robot system. It is able to achieve this runtime because only a subset of the robots are used in a bounded number of phases. The method captures the longest path/geodesic distances in any direction, including the diameter and its associated direction, as well as the average distances of a given MRS. We show theoretical analysis and simulation results with hundreds of robots to validate the algorithm.

REFERENCES