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 $deg(v)$ 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 $\mathcal{Q} : G \to (0, \infty)$ is an algorithm, which on input G, with probability at least 2/3 outputs a value in the interval $[\mathcal{Q}, (1 + \epsilon)\mathcal{Q}]$.

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 2π, 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.

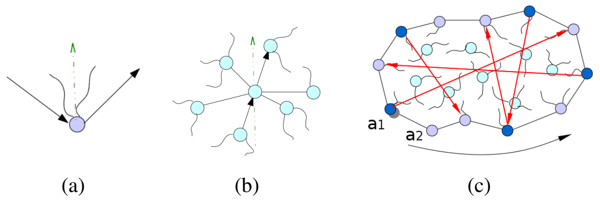
Peripheral Traversal Message (PTM) contains a peripheral robots counter, the probability to trigger the NTM and a short queued history of recently visited peripheral robots.

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.

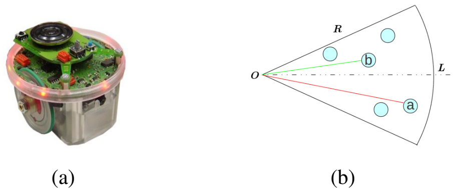
Stage 3: Any robot in S_p immediately passes any received message to its peripheral successor along the determined peripheral "chain." Robot a_1 collects all of the packages and ultimately computes approximate shapes from the two distance metrics.

Stage 4: The results are broadcast via any appropriate propagation strategies to the robots requiring the approx-



(a) (b) (c)

Fig. 1. (a) A periphery agent with the peripheral message passing (solid arrows) and inner "normal" direction (dashed arrow); (b) Message passing between two inner agents along the normal direction; (c) Sampled dark peripheral agents send NTMs. Arrows denote approximate traversal paths.



(a) (b)

Fig. 2. (a) An *e-puck* mobile robot; (b) R-γ-sector with a robot at pole O of polar coordinate system.

imation. These strategies are not the focus of this work and we assume only the initiating robot a_1 requires the approximation.

B. NTM Traversal and R-γ-sector Algorithm

When a NTM is received by a robot, a_i, in S_i, a_i will determine its successor neighbor by selecting the robot nearest to the normal transversal direction. We want to investigate if there is a one-way transversal path which can approximate both of the distance metrics under certain conditions.

In dynamic systems, the edge lengths may not be identical, however each edge length is bounded by a constant sensing/communication radius R. We use a 2-dimensional polar coordinate system to describe the positions of the robots. In this representation, each robot is at the pole O and its goal direction is the polar axis L. Therefore, inside the R-disk, an arbitrary robot a_i can be described with (r_i, θ_i), where $r_i \in [0, R]$, and $\theta_i \in [-\gamma/2, \gamma/2]$. Here, γ 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-γ-sector, see Figure 2(b).

R-γ-sector algorithm: *Inside the R-γ-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-γ-sector, then the neighbor closest to the normal will be chosen.*

We are interested in the constraints of the angle γ 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-γ-sector* algorithm approximates the geodesic distance and shortest path distance well, given there is at least one robot in the *R-γ-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 θ_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-γ-sector* algorithm returns $d_G^{'}$, then we seek a bound in the form

$$d_G^{'} \leq (1+\epsilon)\cdot d_{Gb}, \tag{1}$$

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

$$d_G^{'} = \frac{d_{E0}}{E[r_i^{max}\cdot cos(\theta_i)]} = \frac{d_{E0}}{E[r_i^{max}]\cdot E[cos(\theta_i)]}, \tag{2}$$

where

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

The expectation of the farthest distance in a R-γ-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-γ-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

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

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

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

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

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

Next we want to compare the path distance between the *R-γ-sector* algorithm and the lower bound d_{E0}. The path distance $d_P^{'}$ from our algorithm is

$$d_P^{'} = E[\sum_{i=1}^{\Delta_P^{'}} r_i^{max}] = \sum_{i=1}^{\Delta_P^{'}} E[r_i^{max}] = \Delta_P^{'}\cdot E[r_i^{max}], \tag{7}$$

where $\Delta_P^{'}$ is the number of edges, $\Delta_P^{'} = d_G^{'}$, thus

$$d_P^{'} = \frac{d_{E0}}{E[cos(\theta_i)]}. \tag{8}$$

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

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

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 $q = \Theta(3\sqrt{2}\epsilon^{-2}\sqrt{\tilde{n}})$, where $\tilde{n} = |S_p|$.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Combining (15) and (10) we obtain

$$q \geq 3\epsilon^{-2}(\sqrt{2\tilde{n}+1} - 1). \quad (17)$$

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

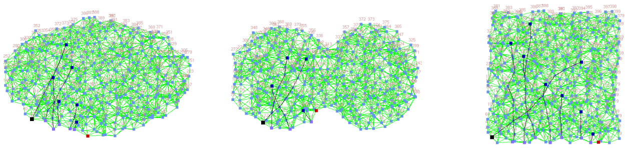
■

D. Running Time Analysis

The overall running time is $O(\tilde{n})$, 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 $O(\tilde{n})$. 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 $O(\tilde{n})$. In Stage 3, there is at most one more round of sequential message passing in S_p, so the running time is still $O(\tilde{n})$, even considering the overhead in Stage 2. The propagation procedure in Stage 4 can also be controlled within $O(\tilde{n})$ by broadcasting from S_p to S_i in parallel). Therefore, the overall running time is $O(\tilde{n})$, 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 $[-5°, 5°]$. 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×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.



(a) (b) (c)

Fig. 3. Typical topological shapes. (a) Convex ellipse with smooth periphery; (b) Non-convex dumbbell with smooth periphery; (c) Convex square with sharp corners.

TABLE I

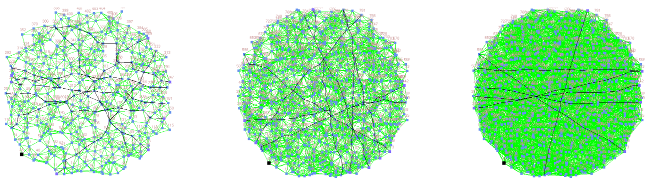
DIAMETER TESTING

Shape	Measure	Increasing degree (fixed $\gamma = 30^\circ$)			Increasing γ (fixed deg=20)		
		deg=10 †	deg=20	deg=30	$\gamma = 40^\circ$	$\gamma = 50^\circ$	$\gamma = 60^\circ$
Ellipse	D_G	17.9	17.1	16.2	15.8	15.1	14.3
	ϵ_{D_G}	0.43	0.37	0.30	0.26	0.21	0.14
	$\sigma_{\angle D_G}$	8.2°	5.0°	3.3°	4.2°	9.7°	18°
	D_P	103	101	99.1	106	104	104
	ϵ_{D_P}	0.03	0.01	0.01	0.06	0.04	0.04
	$\sigma_{\angle D_P}$	10°	7.3°	3.7°	9.8°	3.5°	8.7°
Dumbbell	D_G	17.7	16.9	16.6	16.0	15.0	14.5
	ϵ_{D_G}	0.42	0.35	0.28	0.28	0.20	0.16
	$\sigma_{\angle D_G}$	4.2°	6.0°	3.9°	3.3°	8.2°	11°
	D_P	102	101	99.8	99.5	102	104
	ϵ_{D_P}	0.02	0.01	0.0	0.01	0.02	0.04
	$\sigma_{\angle D_P}$	5.1°	5.7°	2.9°	3.6°	7.0°	9.4°
Square	D_G	16.7	16.0	15.2	15.6	14.9	14.3
	ϵ_{D_G}	0.34	0.28	0.22	0.25	0.19	0.14
	$\sigma_{\angle D_G}$	11°	15°	8.0°	5.8°	17°	11°
	D_P	92	89	93	92	107	112
	ϵ_{D_P}	0.08	0.11	0.07	0.08	0.07	0.12
	$\sigma_{\angle D_P}$	6.5°	11°	21°	9.7°	10.6°	19°

† degree = 10, meaning ∼300 robots in our simulation; degree = 20, meaning ∼600 robots, etc. The sensing range $R = 8m$.

Table I lists the diameter testing results of the three typical shapes. The measurements include the geodesic diameter D_G, the ϵ value ϵ_{D_G} with regard to lower bound D_{Gb}, and the standard deviation of angles $\sigma_{\angle D_G}$ indicating deviation from the true diameter directions. Similarly, D_P, ϵ_{D_P} and $\sigma_{\angle D_P}$ are the counterparts of shortest path distances. We first fixed the γ 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 γ 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 γ 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-γ-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 γ 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).



(a) (b) (c)

Fig. 4. Quality of traversal paths with different degrees. (a) $deg \approx 12$ (n=400, R=8); (b) $deg \approx 24$ (n=800, R=8); (c) $deg \approx 48$ (n=800, R=12).

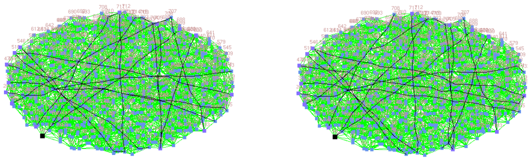
TABLE II

R-γ-SECTOR ($R\gamma$) VS. NORMAL CLOSEST (NC)

	Degree	10	15	20	25	30
Ellipse	$NC\ D_G$	17.2	17.8	18.3	18.5	19.1
	$R\gamma\ D_G$	17.9	17.2	17.1	17.0	16.2
	$NC\ D_P$	101	100	101	99.9	99.7
	$R\gamma\ D_P$	103	102	101	101	99.1
Dumbbell	$NC\ D_G$	16.7	17.2	18.0	18.4	18.8
	$R\gamma\ D_G$	17.7	17.7	16.9	16.6	16.1
	$NC\ D_P$	99.9	99.6	99.5	100	99.2
	$R\gamma\ D_P$	102	104	101	99.7	99.8
Square	$NC\ D_G$	15.9	16.6	17.6	17.6	18.4
	$R\gamma\ D_G$	16.7	16.1	16.0	15.7	15.2
	$NC\ D_P$	85.2	87.0	95.5	93.1	91.0
	$R\gamma\ D_P$	92.1	89.9	89.9	92.2	93.4

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 ϵ value. The results show, as ϵ 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 ϵ increases, the average distances $\bar{d_G}$ and $\bar{d_P}$ oscillate with increasing amplitude and tend to diverge, which is indicated by the standard deviations σ_{d_G} and σ_{d_P}, 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



(a) (b)

Fig. 5. Paths when degree ≈ 20. (a) Path of normal-closest method; (b) Path of R-γ-sector method (γ=30°).

TABLE III

BOUNDING NUMBER OF NTM SAMPLES

Shape	Measure	ϵ=0.3	ϵ=0.6	ϵ=0.9	ϵ=1.2	ϵ=1.5	ϵ=1.8
Ellipse	P_s	1.0	1.0	0.71	0.40	0.25	0.17
	D_G	15.2	15.2	14.5	14.1	14.0	13.6
	D_P	101	101	98.6	98.2	98.5	95.4
	$\bar{d_G}$ (σ_{d_G})	10.3 (0.2)	10.3 (0.2)	10.3 (0.4)	10.0 (0.5)	10.5 (0.8)	10.7 (1.1)
	$\bar{d_P}$ (σ_{d_P})	68.7 (2.9)	68.7 (2.9)	67.6 (5.1)	67.7 (7.7)	70.4 (8.0)	69.0 (11.4)
Dumbbell	P_s	1.0	1.0	0.71	0.40	0.25	0.18
	D_G	15.9	15.9	15.2	13.7	14.0	12.3
	D_P	102	102	100	99.7	98.3	90.9
	$\bar{d_G}$ (σ_{d_G})	9.7 (0.5)	9.7 (0.5)	10.4 (0.8)	9.5 (0.7)	9.4 (1.2)	8.9 (1.7)
	$\bar{d_P}$ (σ_{d_P})	64.3 (3.5)	64.3 (3.5)	63.6 (6.2)	62.1 (8.2)	67.8 (7.4)	59.5 (9.3)
Square	P_s	1.0	1.0	0.68	0.38	0.25	0.17
	D_G	14.4	14.4	13.6	12.9	12.2	11.7
	D_P	94.9	94.9	92.1	88.4	78.0	80.4
	$\bar{d_G}$ (σ_{d_G})	10.2 (0.3)	10.2 (0.3)	10.0 (0.9)	10.7 (2.1)	10.3 (1.7)	9.2 (2.8)
	$\bar{d_P}$ (σ_{d_P})	70.4 (4.1)	70.4 (4.1)	71.2 (7.5)	71.8 (7.3)	72.0 (9.8)	66.6 (15.2)

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

TABLE IV

TIME COMPLEXITY TESTING WITH 1000 ROBOTS

Radius	Measurement	Circle	Ellipse	Dumbbell	Square
$R = 8m$	$\vert S_p \vert$	65	54	55	59
(deg≈30)	$time\ steps$	187	157	162	170
$R = 6m$	$\vert S_p \vert$	87	70	74	80
(deg≈15)	$time\ steps$	246	203	214	226

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

[1] O. Goldreich and D. Ron, "Approximating average parameters of graphs," *Electronic Colloquium on Computational Complexity (ECCC)*, no. Report No. 73, 2005.

[2] B. Chazelle, D. Liu, and A. Magen, "Sublinear geometric algorithms," *SIAM Journal on Computing*, vol. 35, no. 3, pp. 627–646, Mar. 2006.

[3] P. Indyk, "Sublinear time algorithms for metric space problems," *Proceedings of the 31st Annual ACM Symposium on Theory of Computing (STOC)*, pp. 428–434, 1999.

[4] O. Goldreich, "Combinatorial property testing (a survey)," *Proceedings of the DIMACS Workshop on Randomization Methods in Algorithm Design*, vol. 43, pp. 45–59, 1997.

[5] A. Czumaj, C. Sohler, and M. Ziegler, "Property testing in computational geometry," *Proc. 8th ESA*, pp. 155–166, 2000.

[6] C. A. Duncan, M. T. Goodrich, and E. A. Ramos, "Efficient approximation and optimization algorithms for computational metrology," *Proc. 8th ACM-SIAM SODA*, pp. 121–130, 1997.

[7] O. Goldreich and D. Ron, "Property testing in bounded degree graphs," in *STOC '97: Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, 1997, pp. 406–415.

[8] O. Goldreich, "Property testing in massive graphs," in *Handbook of massive data sets*, 2002, pp. 123–147.

[9] C. M. Cianci, X. Raemy, J. Pugh, A. Martinoli, and E. P. Federaledelausanne, "Communication in a swarm of miniature robots: The e-puck as an educational tool for swarm robotics," in *Simulation of Adaptive Behavior, Swarm Robotics Workshop*, 2006, pp. 103–115.

[10] A. Gutirrez, A. Campo, M. Dorigo, D. Amor, L. Magdalena, and F. Monasterio-Huelin, "An Open Localization and Local Communication Embodied Sensor," *Sensors 2008*, pp. 7545–7563, Aug. 2008.

[11] A. Gutirrez, A. Campo, and M. Dorigo, "Open E-puck Range & Bearing Miniaturized Board for Local Communication in Swarm Robotics," *2009 IEEE International Conference on Robotics and Automation*, May 2009.