An Efficient Distributed Topo-Geometric Spatial Density Estimation Method for Multi-Robot Systems

Lantao Liu and Dylan A. Shell

***Abstract*— A fundamental challenge in multi-robot systems is that global information is needed to succeed in some tasks, while the system's computation and sensing are fundamentally distributed. This paper considers the problem of estimating the relative density of robots in particular regions of the environment, but without wishing to incur the cost of obtaining a consistent metric representation. We compute a probability density function that describes positions of the robots within the system by leveraging properties of the underlying communication network. We introduce three different strategies for using and combining local measurements via a modified Parzen window kernel density method. The result is a representation that is most accurate near to the querying robot but which maintains qualitative properties of the global density. We argue that this a useful relaxation of the problem because it is meaningful from the perspective of the robots within the system itself. Validation takes the form of simulations with hundreds of simple robots.**

I. Introduction

A multi-robot team has the potential to outperform a single robot through the synergy of the efforts and capabilities of the constituent robots. But in many situations, even when the computation is not centralized, decision making still depends on knowledge of the whole system. We are particularly interested in collecting statistical descriptions of system wide properties in order to aid in performing these types of tasks. This work considers the problem of estimating the spatial density of a completely distributed multi-robot system.

Density estimation is a technique for constructing an estimate of an unobservable underlying probability density function (p.d.f.) based on limited observations. The way mobile robots are spread throughout a space can be modelled as a density function over Cartesian coordinates. Density estimation is traditionally used to visualize and understand the structures and patterns underlying some data. While this is usually done with conventional metrics, in this work we explore algorithms that provide density estimates that are topo-geometric (first coined in [1]), bringing together complementary aspects of both global topological and geometrical properties for large scale networks. We posit that such metrics are particularly useful for robots autonomously making decisions in a "fog" of uncertainty. Topo-geometric methods use a standard metric locally to provide accurate measurement in neighborhood around each robot, but which are more qualitative further out. Because "globally topo-geometric, locally metric" densities can be meaningfully employed by the robots in reasoning about their teammates, it is useful to apply techniques that allow one to estimate such densities in a distributed way.

Lantao Liu and Dylan A. Shell are with Dept. of Computer Science and Engineering, Texas A&M University, College Station, TX 77843, USA. {lantao, dshell}@cse.tamu.edu

Although a variety of methods of density estimation over distributed data already exist [2], [3], [4], [5], [6], global density estimation for the specialized problem of estimating the spatial positions for distributed mobile robots has not yet been addressed adequately. In particular, existing methods, among which most are related to sensor networks, remain ill-suited because they make assumptions which do not reflect reality in a multi-robot system. Distinctions of density estimations between distributed multi-robot systems and sensor networks are described and compared in Section III.

We develop one such technique in this paper: the global p.d.f. is estimated by using non-parametric statistics—namely, the Parzen window density estimation method. Inspired by this method, communication is used to collect measurements for each sample and emulate each window (the windows, for which the method gets its name, are usually centered on data points). Local observations are aggregated via a tree structure through which the data flow. Both global and local spatial features are captured by the combination of the data-flow tree and the local observations. In addition, time complexity (running iterations) and communication complexity (total number of end-to-end connections) are reduced by fusing data on the compact data-flow tree.

The proposed method has dual features: it is locally metric but globally topo-geometric. In essence, *metric consistency is achieved exactly only locally* (*i.e.*, the usual properties are maintained in a local patchwork represented by the communications network), *while the topo-geometric aspect identifies global structures by linking the local patchworks together* (see also [1]). The idea that one gains from complementary topological and metric representations has been use in several ways in robotics including mapping and localization [7], [8], coverage control [9] and motion planning [10]. We are not aware of any work, however, for employing topological properties with a view to density estimation. Conventional density estimation employs traditional metrics and the very definition of density is meaningful over a space (*e.g.*, with well-defined metric). In contrast, a deformed density function in the topological sense can be appropriately and meaningfully interpreted for the robot's themselves. This is because these operations use exactly the same connectedness relation (*e.g.*, the same data-flow tree). Our primary motivation lies in that such estimated densities can be effectively used for operations such as the density-based clustering or partitioning, as well as gradient-based redistribution of the distributed multi-robot systems, *etc*.

II. Problem Description

A. Parzen Window Density Estimation

The goal of density estimation is to make inferences about the p.d.f. underlying a finite set of data samples, despite the fact that there are locations where no data are observed. We employ a popular non-parametric kernel density estimation (KDE) technique called the Parzen window method. KDE is a data-driven approach where the contribution of each data point is smoothed out from a single sample point into a region of space surrounding it. Aggregating the smoothed contributions yields a p.d.f. which reflects the structure of the data.

More formally, given a set of d-variate data samples $\{\mathbf{X}_i, i = 1, \cdots, N, \mathbf{X}_i \in \mathbb{R}^d\}$ drawn from a density f, the kernel density estimate for this set is:

$$\hat{f}(\mathbf{x}, \mathbf{H}) = \frac{1}{N} \sum_{i=1}^{N} \phi(\mathbf{x}, \mathbf{X}_i; \mathbf{H}), \tag{1}$$

where $\mathbf{x} = (x_1, x_2, \cdots, x_d)^T$ and $\mathbf{X}_i = (X_{i1}, X_{i2}, \cdots, X_{id})^T$, $i = 1, 2, \cdots, N$. $\mathbf{H}$ is symmetric and positive-definite and termed the *bandwidth* matrix, which is used for smoothing and scaling purposes. $\phi(\mathbf{x}, \mathbf{X}_i; \mathbf{H})$ is a *kernel* with location $\mathbf{X}_i$ and $\mathbf{H}$:

$$\phi(\mathbf{x}, \mathbf{X}_i; \mathbf{H}) = |\mathbf{H}|^{-1} \kappa(\mathbf{H}^{-1}(\mathbf{x} - \mathbf{X}_i)), \tag{2}$$

where $\kappa(\mathbf{x})$ is a symmetric probability density function called *kernel function*. Specific choices for kernel functions are not critical to our considerations here, so long as the contributions of individual sample points are smoothed. In our work we use $\kappa \sim \mathbf{N}_d(\mathbf{0}, \mathbf{I})$, namely:

$$\kappa(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}} \exp(-\frac{1}{2}\mathbf{x}^T\mathbf{x}). \tag{3}$$

Let $\mathbf{\Sigma}$ be the covariance matrix of vector $\mathbf{x}$. We consider the simple and common case, where $\mathbf{H}$ and $\mathbf{\Sigma}$ are restricted to diagonal matrices, *i.e.*, $\mathbf{H} = \text{diag}(h_1, h_2, \cdots, h_d)$ and $\mathbf{\Sigma} = \text{diag}(\sigma_1^2, \sigma_2^2, \cdots, \sigma_d^2)$, then the rule of thumb (Scott's rule [11]) for choosing the bandwidth is

$$h_j \doteq \sigma_j \left(\frac{4}{(d+2)N} \right)^{1/(d+4)} \quad j = 1, 2, \cdots, d. \tag{4}$$

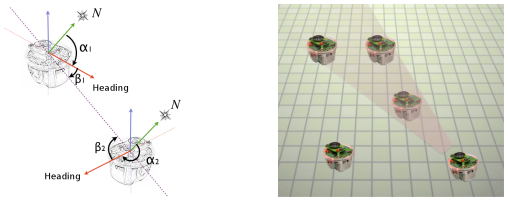
We are interested in estimating the spatial density function, so the data samples are *bivariate* vectors, *i.e.*, $\mathbf{X}_i = (x_i, y_i)^T$, where x_i, y_i are the coordinate values of the i-th robot's position. Following Equation (4), the bandwidth estimate along the x, y directions simplifies to

$$\hat{h}_x = \hat{\sigma}_x N^{-1/6}, \quad \hat{h}_y = \hat{\sigma}_y N^{-1/6}. \tag{5}$$

The concept of Parzen window suggests that one might measure the local observations within a "window" with circular boundary. Let radius R_w of the window be

$$R_w = K \cdot \max(\hat{h}_x, \hat{h}_y), \tag{6}$$

where K is an empirically determined parameter depending on the quality of final estimated density functions. It is difficult to derive a theoretical K because the bandwidths h_x and h_y are computed via the Scott's rule which is empirical. Further discussion of K is given in experimental section.



(a) (b)

Fig. 1. (a) N represents the north direction sensed by the compass, and red arrows denote the robot heading directions relative to N (assuming clockwise angular direction). Absolute bearing of a neighbor can be calculated by $\alpha + \beta$; (b) an illustration of IR signal occlusion.

B. Neighbors: Sensing, Measurement and Communication

We assume the robots are homogeneous and without any centralized controller. The sensing information that we use for spatial density estimation is limited: distances and bearings to neighboring robots. This can be obtained via an array of ranging sensors such as IR or sonar. A compass is used to compute absolute bearing information (see Fig. 1(a) for an illustration). Robots are also required to be able to communicate with their neighbors. These requirements are minimal so as to be cheap and reflect existing hardware; for example, e-puck robot with an *e-RandB* (range and bearing miniature board [12]) is available and permits the robots to identify, detect and communicate with a small subset of nearby robots.

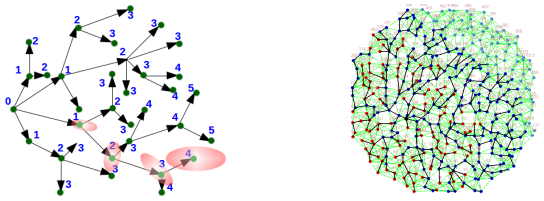
All local communications are treated as message passing; sent messages are received by all the robots within the communication range, and a communication channel is established after a handshake by verifying the identifiers and/or the computed absolute bearings of nearby robots. Occlusion poses a challenge. Since our method requires the spatial information of all robots within the Parzen window, the accuracy of the estimate decreases as robots within the window are occluded. Fig. 1(b) shows how two robots (at the top of figure) are blocked by the one in the center, rendering them invisible to the robot in the lower-right corner. Such occlusions are dealt with at the slight cost of increased communication.

We assume each robot has bidirectional communication and model the system as a sparse undirected graph $G = (V, E)$, where the vertices V represent the positions of the robots ($|V| = n$), and the edges E represent the connections of directly communicable robot pairs ($e(v, u) \in E$ is an edge between vertex $v \in V$ and $u \in V$). Two parameters that are useful to us are: (1.) the *degree* of a vertex $deg(v)$, which is the number of edges connected to this node; (2.) the *diameter* of the graph $diam(G)$, that measures the maximum distance[1] between any two nodes in G.

III. Local Observations, Global Combination

Unlike the original Parzen window method, where the global density function is an average of kernelized densities drawn from all samples, our method considers local observations of each sample (robot), and all local observations contribute to correct the measurement errors that deform the

[1]The distance here means the geodesic distance, *i.e.*, the minimal number of edges between two vertices in a graph. See [13], [14] for applications of diameter in multi-robot systems.



(a) (b)

Fig. 2. (a) A data-flow tree showing messages traversals. Numbers on the nodes denote the hop distance from the root. Bigger ellipses indicate larger estimation errors; (b) Messages traversing the network. Light blue nodes are unexplored whereas dark blue nodes are recently explored. Red nodes denote visits by DT-Msgs in backward traversals starting from leaf nodes.

p.d.f. Spatial density estimation of the multi-robot systems has a unique set of challenges:

1) The network may be incomplete. The robots can only communicate with others within their vicinity;
2) Data are not ready-made. An important part of the problem involves the robot performing the measurement itself, which it must do before the results can be sent to the data collectors;
3) Communication costs are significant and should be minimized. Communication reductions produces a more responsive system.
4) Data have measurement errors. Additionally, errors may accumulate or be magnified as part of the data collection procedure.

The following subsections provide the solutions to adapt to these particular characteristics.

A. Data Flow Tree—Addressing the Incomplete Network

To estimate the global density, the broad perspective we employ is the *local estimations/observations, global combination* framework already used elsewhere for distributed systems (see [4], [5] for examples). Unlike existing methods within the framework, we design a communication protocol based on a tree-like structure which aids in managing the flow of messages. We call the tree a *data-flow tree (DFT)*, where the query robot at the root represents a "sink" and the other nodes are sample locations, see Fig. 2(a).

A DFT is constructed by identifying the parent-children relationship. The *DFT-construction message* (DC-Msg) makes this possible. A DC-Msg package contains the following:

- the preceding (parent) robot's ID and derived absolute position;
- the hopping distance from the root;
- a short queued history of recently visited robots.

A DFT is constructed only upon request of a query robot (for our purposes, an arbitrary robot). The query robot acts as the root, initiates multiple DC-Msg packages, and distribute them to a subset of the nearest neighbors that are directly communicable. A robot that has received a DC-Msg immediately flags its state as having been explored and will not accept other DC-Msgs. All such robots update their DC-Msg, replicate it, and distribute it to other robots in the same way. Successive robots naturally have larger numerical labels, each being identical to the distance (in hops) from the root, as shown in Fig. 2(a). A DC-Msg ends its life when it reaches a leaf node (no other neighbors can be explored), and a DFT is fully constructed once all DC-Msgs have disappeared.

One detail worth mentioning is that not all communicable neighbors are passed DC-Msgs. Instead, we predefine a probability $p \in (0, 1]$ to control the degree of DFT nodes. This means that for a node v, if $deg(v) > 1$, it distributes the DC-Msg to at most $\lceil p \cdot deg(v) \rceil$ of the nearest communicable neighbors (which also depends on the availability of unexplored nodes). The remaining edges in the communication graph but not in DFT are used for data fusion, which is discussed in following subsections.

Once the DFT has been established, a *DFT-traversal message* (DT-Msg) can be used to convey data along the DFT branches, either in a *forward* (away from the root) or *backward* (toward the root) direction. The differences between DT-Msg and DC-Msg are: (1.) a DT-Msg does not have a limited lifetime and contains extra state recording the current traversal direction; (2.) at each node with more than one child, DT-Msgs can be replicated in a forward traversal as well as merged in a backward traversal.

After all the data have been collected by DT-Msgs, the query robot then computes the global p.d.f., and, if necessary, may broadcast it to the whole system. We call the process in which a DFT is completely traversed both forward and backward a single *round*.

B. Local Measurement—Collecting the Data

During the construction of the DFT, the robots are linked together. A DC-Msg records the absolute coordinate $\vec{P}_{0,a}$ (relative to root 0) of robot a which sent it out, and the successor b that receives this message is able to derive its absolute coordinate $\vec{P}_{0,b}$ based on the relative coordinate $\vec{P}_{a,b}$ between a and b (associated with immediate range and bearing observations):

$$\vec{P}_{0,b} = \vec{P}_{0,a} + \vec{P}_{a,b}, \tag{7}$$

thus, if we fix the absolute position of the root of the tree, all other robots at hopping distance m in a chain can be localized along the DFT:

$$\vec{P}_{0,m} = \sum_{i=0}^{m-1} \vec{P}_{i,i+1}, \quad \forall m \geq 1. \tag{8}$$

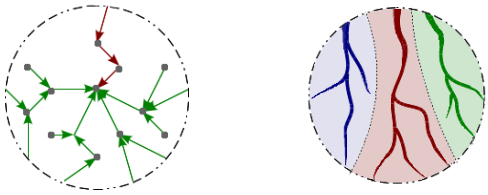
Once a robot receives a DC-Msg and flags its state as in the DFT tree, it needs to prepare the data (local observations) that will be collected by a backward traversing DT-Msg. We propose three strategies to obtain such local statistics:

1) No Local Metric: Each robot simply reports its own absolute position that it estimated along the message traversals of DFT construction. For a system with n robots, the root will finally get n data samples. This is a traditional "local estimation, global combination" method to collect distributed data and works well when sensing error is small. In essence, this strategy uses only the "connectedness" of the network topology; it is cheap in terms of communication messages and fast in running time.

2) Coarse Local Metric: We can extend the strategy above if we add a little extra local measurement: *i.e.*, each robot i considers the neighbors j ($j \neq i$) it can sense locally to form a window (with radius at most the sensing range R_s) instead of only itself. These neighbors form a set S:

$$S = \{j \mid \| \mathbf{X}_i - \mathbf{X}_j \| \leq R_s\}, \quad \forall \text{ sensible } j. \tag{9}$$

Explicit communication with these neighbors is not necessary, instead, their positions are derived with observations of



(a) (b)

Fig. 3. (a) A local DFT. Red branch is part of the parent branch and is obtained from forward DT-Msg traversal. Green branches are part of children branches and are obtained from backward DT-Msg traversals; (b) Fusion of local observations from different incomplete local DFTs.

their relative ranges and bearings. This strategy requires extra observations and computation, but no extra communication.

3) Fine Local Metric: We add extra communication to overcome occlusion, allowing a robot to "see through" the obstacles and include measurements for robots that are farther away, *i.e.*, observations of neighboring robots are combined via communication. This local exploration is achieved by building a miniatured DFT rooted at the measuring robot i which only grows as large as the border of the Parzen window (with radius R_w). The set of robots on such miniatured DFT can be expressed as:

$$S = \{j \mid \| \mathbf{X}_i - \mathbf{X}_j \| \leq R_w\}, \quad \forall j \text{ on DFT}. \tag{10}$$

The above three strategies are given in order of dependence on metric information to nearby agents (and therefore also communication). The locally measured data are then aggregated and "kernelized" by the root robot in the final stage and a p.d.f is computed by considering all local observations. Mathematically, a sample (robot) X_i can be observed in different windows, with observed values $\hat{X}_i^{(j)}$ in the j-th window subject to robot j. Let J_i denote the set of robots who can observe X_i, then $J_i = \{j\}$, and a local p.d.f. contributed by X_i is averaged from all these observations:

$$\hat{f}(\mathbf{x}, \hat{\mathbf{X}}_i) = \frac{1}{|J_i|} \sum_{\forall j \in J_i} \phi(\mathbf{x}, \hat{\mathbf{X}}_i^{(j)}; \mathbf{H}), \tag{11}$$

where the kernel for an observation is

$$\phi(\mathbf{x}, \hat{\mathbf{X}}_i^{(j)}; \mathbf{H}) = |\mathbf{H}|^{-1} \kappa(\mathbf{H}^{-1}(\mathbf{x} - \hat{\mathbf{X}}_i^{(j)})), \ \forall j \in J_i, \tag{12}$$

and the final p.d.f. becomes

$$\hat{f}(\mathbf{x}, \mathbf{H}) = \frac{1}{N} \sum_{i=1}^{N} \hat{f}(\mathbf{x}, \hat{\mathbf{X}}_i), \tag{13}$$

which in essence is averaged from all estimated kernels.

C. DFT Data Fusion—To Reduce Communication

As mentioned in Subsection III-B, a good local measurement for a data sample can be obtained by constructing a local miniatured DFT rooted at that sample. However, since many other robots—including the neighboring robots—are performing exactly the same work simultaneously, this can cause communication traffic conflicts. Even if the conflicts can be alleviated through serialization, it remains possible for a communication edge to be used many times to convey the same information.

To solve this problem, we implicitly construct such miniature DFTs by selectively recording those local measurements along with the DT-Msg passing in one round of DFT traversals. More specifically, in the forward traversal, the message queues a short history of the preceding robots' information, such that a parent branch is used to serve as a tree-branch of local DFT (see the red branch in Fig. 3(a)). Similarly, histories of the child branches can be recorded in the backward traversal (green branches in Fig. 3(a)).

However, the method described will not necessarily capture all the robots in the Parzen window, *e.g.*, in Fig. 3(b), there are three independent branches within the window, and none of them can contain the correct local measurements alone. To solve this problem, we employ a technique to "fuse" the data by constructing the union of sets. The robot updates its local DFT by detecting and adding new robots within the window. A measuring robot $i \in S$ communicates with the neighbors $j \in S^{'}$ that are not in the same branch and exchanges the information on related robots (S and $S^{'}$ are sets from different local DFTs). The spatial positions for the set of exchanged robots are re-derived and the ones that are inside a Parzen window are incrementally added to the local DFT, *i.e.*,

$$S = S \cup \{j \mid \| \mathbf{X}_i - \mathbf{X}_j \| \leq R_w\}, \quad \forall j \in S^{'}. \tag{14}$$

Since a robot (*e.g.*, node v) has at least $(1-p) \cdot deg(v)$ local connections that are not on DFT, approximately $(1-p) \cdot deg(v)$ times this fusion will be successful. In this way, the utility of each communication edge can be maximized: conflict need not happen since the fusion communications over the off-DFT edges are (pairwise) single hops.

The high-level pseudo-code is summarized in Algorithm IV.1. The algorithm uses two rounds of DFT traversals: the first round is responsible for constructing the DFT and collecting the N data points without any local metric measurements. The coarse data are used to approximate bandwidth with the rule of thumb; then the second round using specific local metrics further refines the estimation.

IV. Experiments

In order to validate the approach, we simulated the algorithm with hundreds of robots on our simulator written in C++ and MATLAB. All the robots are homogeneous and have identical sensing and communication ranges. Each robot is capable of recognizing its neighbors if they are within the sensing/communication range and there is no other obstacle in between (robots do occlude communications of others). For these recognizable neighbors, communication edges are established with them, and their ranges and bearings can be directly observed. We also simulate measurement errors, and perturb each neighbor's sensed distance by a random value in the range $[-50cm, 50cm]$ and perturb every neighbor's sensed bearing by a random value in the range $[-30°, 30°]$.

Fig. 4(a) is an example of a multi-robot swarm ($N = 500$) carrying out tasks under two Gaussian distributions (practical scenarios can be, for instance, marine surface robots cleaning oil or a red tide formed with measurable distributions). Along with the task commitment, the positions of robots may drift and the distribution of the whole swarm can change over time. Therefore, immediate estimation of the global p.d.f. based on real-time measurements is required to monitor the system and adapt the densities of robots. In order to best

Algorithm IV.1 Spatial density estimation algorithm

Input: capabilities of local localization and communication
Output: global p.d.f.

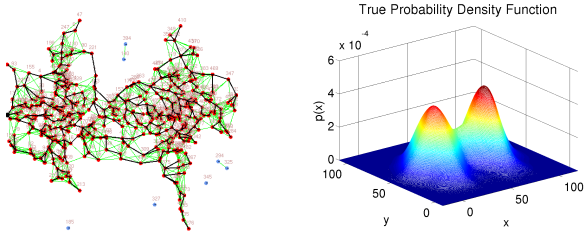
{/* 1st round traversal */};
1: DFT construction (forward);
2: coarse data aggregation (backward);

{/* at the root */};
3: bandwidth computation using rule of thumb;

{/* 2nd round traversal */};
4: obtaining parent branches of local DFTs (forward)
5: obtaining children branches of local DFTs (backward)
6: data fusion to perfect local DFTs (backward)
7: aggregation of local observations (backward)

{/* at the root */};
8: p.d.f. computation by query robot.

Notes: Lines 4—6 can be substituted with coarse metric.



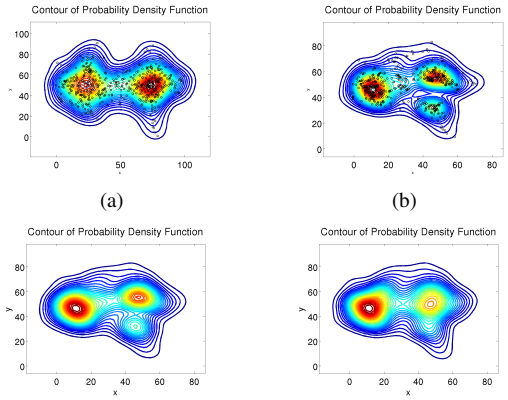
(a) (b)

Fig. 4. An example of a 2-peak Gaussian-distributed swarm in an 100m×100m planar space (a) The constructed DFT with root on the left (worst case for estimation). Red circles denote robots; (b) The true probability density function estimated from the ground truth coordinates.

estimate the global density, we require the majority of the robots be connected. To demonstrate the worst case, the query robot is chosen from the perimeter of the network, as the big black nodes on the far left. Fig. 4(b) is the estimated p.d.f. based on the ground truth positions.

To compare the three local metrics, contours from each strategy are plotted, as shown in Fig. 5. Fig. 5(a) is the contour generated from the ground truth positions, and the other three are the ones that use no local metric, coarse local metric and fine local metric, respectively. Comparing the four contours, we can see that none of the last three are identical in shape or in scale to that of the ground truth. However, if we count the number of peaks, only Fig. 5(d) (the one that uses fine local metric) has the same number as that of the ground truth, although the contour shape of the right Gaussian is wider than the left. Intuitively, one can imagine that the right Gaussian contour in Fig. 5(d) is the stretched counterpart of Fig. 5(a). The shape is deformed, but it is not split into pieces. Such a property is conceptually similar to the notion of topological deformation. Thus, the "fine local metric" resolves issues arising in the first two local measurement strategies in this case.

One thing worth mentioning is that, as the number of samples in the same space increases, the size of Parzen window decreases. When the Parzen window is smaller than the robot's sensing/communication range, the coarse local metric is able to cover the area to take into account enough neighbors. In that case, estimation using the local coarse



(a) (b) (c) (d)

Fig. 5. Contours from differing local metrics. (a) Ground truth (No Measure); (b) No metric; (c) Coarse local metric; (d) Fine local metric.

metric can reach the same accuracy as that from the fine local metric, which is illustrated in Fig. 6.

It is difficult to find a meaningful distance metric to compare our p.d.f. with that of the ground truth (see a survey for possible distance metrics in [15]), since the data are deformed (errors are embedded) before the estimation step.

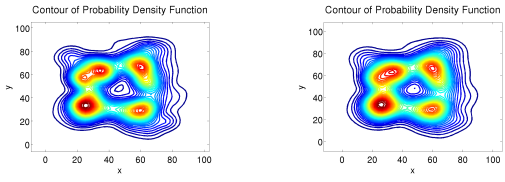
One possible way to quantify the differences among these deformed density functions is to compare the p.d.f. values of the original samples (robots' positions): $\hat{f}(\mathbf{X})$, where $\mathbf{X} = \{X_i, i = 1, \cdots, N\}$. The rationale is that, although the spatial positions and scale are changed due to measurement errors, the topology is expected to be unchanged (as the two peaks vs. three peaks in Fig. 5). If we locate the samples in the corresponding deformed p.d.f.'s, they should have similar values. However, such sampling on a p.d.f. is not very meaningful since the samples are finite, but the p.d.f. is continuous. In a sense this form of comparison is akin to comparison of two vectors, nevertheless, some sense of similarity between the two estimations can be grasped. We use three distance metrics from differing families: Euclidean distance d_E from the L_p Minkowski family, Sorensen distance d_S from the L_1 family, and Fidelity distance d_F from the Fidelity family, as below:

$$d_E = \sqrt{\sum_{i=1}^{N} (\hat{f}_1(X_i) - \hat{f}_2(X_i))^2}, \tag{15}$$

$$d_S = \frac{\sum_{i=1}^{N} |\hat{f}_1(X_i) - \hat{f}_2(X_i)|}{\sum_{i=1}^{N} (\hat{f}_1(X_i) + \hat{f}_2(X_i))} \tag{16}$$

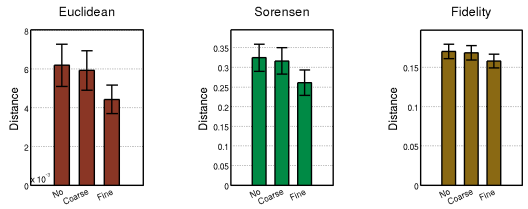
$$d_F = \sum_{i=1}^{N} \sqrt{\hat{f}_1(X_i)\hat{f}_2(X_i)} \tag{17}$$

We set $\hat{f}_1(\cdot)$ to the ground truth density function, and $\hat{f}_2(\cdot)$ to the density functions to be compared, which can be the estimated p.d.f.'s of using no local metric, coarse



(a) (b)

Fig. 6. Spatial density estimations for a large system with 1000 robots distributed in four Gaussian clouds (query robot is near the coordinate origin). (a) Coarse local metric; (b) Fine local metric.



(a) (b) (c)

Fig. 7. Similarities analysis using different distance metrics. (a) Euclidean; (b) Sorensen; (c) Fidelity.

local metric, or fine local metric. Fig. 7 shows the results for testing the scenario of two Gaussian peaks formed by 500 robots, using the three local measurement methods, each of which is compared with the ground truth with the three distance metrics of (15)– (17). All three distance metrics show that the measurements of fine local metric are much better than the methods with no or coarse local metrics.

Similar to the bandwidth computation in various Parzen window approaches, our window radius R_w is also an empirical value. As described in Equation (6), we need to determine the constant K. We have investigated the influences of window radius with the mentioned three distance metrics, as summarized in Fig. 8. The three curves have a consistent trend: as the radius of our Parzen window increases, the estimated p.d.f. is closer to the truth. This reflects the trade-off between the computation/communication and the fineness of estimated density function. We tested various cases with different numbers of peaks, and the results show that when R_w is $1 \sim 2$ times the bandwidth ($\max(h_x, h_y)$), the global density features can be clearly captured and corrected. This corresponds to the "radii bandwidth" of 8–16 along x-axis of sub-figures in Fig. 8. For the window radii larger than 16, the distance changes tend to flatten out. Therefore, to be conservative, K should be set to be 2 for an unknown set of samples.

V. Discussion and Conclusion

Both time complexity and communication complexity are important for the applications we envision. The time complexity is measured by the time steps for message passing hops. There are two rounds of message traversals, in each round the running time is $O(diam(G))$, where $diam(G)$ can be approximated with $O(\log N)$. Therefore, the overall running time is approximately $O(\log N)$ on average. For instance, in our experiment the running time for estimating a system of $N = 1000$ is only ~ 200 time steps. The worst case for this method costs $O(N)$ time steps, when all robots are arranged in a straight line. The communication connectivity complexity is $O(|E|)$: the messages traversing along the DFT only use the edges in the tree while the data fusion steps use the remaining edges.

This paper introduces an effective algorithm to estimate the p.d.f. of mobile robots' spatial positions. The density estimation utilizes two complementary parts: a measure captures the detailed metric features of the local neighborhood, an a global topo-geometric measure characterizes the overall large scale structure. The deformed p.d.f in the topological sense captures density information that can be well utilized in other operations which use the same connectedness relation (*e.g.,*

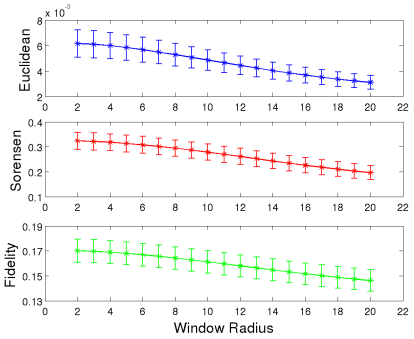


Fig. 8. The distances between the estimated density and the ground truth density. y-axis is the distances, and x-axis is the window radius for local metric. Each result is averaged from 10 experiments.

partitioning, gradient-based system redistribution). Finally, we verified the algorithm within a large-scale multi-robot simulation and interpreted both quantitative and qualitative aspects of the results.

References

[1] B. Hillier, A. Turner, T. Yang, and H. T. Park, "Metric and Topo-Geometric Properties of Urban Street Networks: Some Convergence, Divergence, and New Results," *Journal of Space Syntax*, vol. 1, no. 2, pp. 258–279, 2010.

[2] R. Nowak, "Distributed EM Algorithms for Density Estimation and Clustering in Sensor Networks," *Signal Processing, IEEE Transactions on*, vol. 51, no. 8, pp. 2245 – 2253, aug. 2003.

[3] B. Safarinejadian and M. B. Menhaj, "Distributed Density Estimation in Sensor Networks based on Variational Approximations," *International Journal of Systems Science, 42(9),*, pp. 1445–1457, Sept. 2011.

[4] M. Klusch, S. Lodi, and G. Moro, "Distributed Clustering Based on Sampling Local Density Estimates," in *Proceedings of the Joint International Conference on AI (IJCAI)*, 2003.

[5] C. Giannella, H. Dutta, S. Mukherjee, and H. Kargupta, "Efficient Kernel Density Estimation over Distributed Data," in *The 9th International Workshop on High Performance and Distributed Mining, SIAM International Conference on Data Mining*, 2006.

[6] Y. Hu, H. Chen, J.-g. Lou, and J. Li, "Distributed Density Estimation Using Non-parametric Statistics," in *Proceedings of the International Conference on Distributed Computing Systems (ICDCS'07)*, 2007.

[7] S. Thrun, J. steffen Gutmann, D. Fox, W. Burgard, and B. J. Kuipers, "Integrating Topological and Metric Maps for Mobile Robot Navigation: A statistical Approach," in *In Proceedings of the AAAI Fifteenth National Conference on Artificial Intelligence*, 1998.

[8] P. Newman, G. Sibley, M. Smith, M. Cummins, A. Harrison, C. Mei, I. Posner, R. Shade, D. Schroeter, L. Murphy, W. Churchill, D. Cole, and I. Reid, "Navigating, Recognizing and Describing Urban Spaces With Vision and Lasers," *International Journal of Robotics Research*, vol. 28, no. 11-12, pp. 1406–1433, November 2009.

[9] J. Derenick, V. Kumar, and A. Jadbabaie, "Towards Simplicial Coverage Repair for Mobile Robot Teams," in *IEEE International Conference on Robots and Automation (ICRA)*, 2010.

[10] M. Farber, S. Tabachnikov, and S. Yuzvinsky, "Topological Robotics: Motion Planning in Projective Spaces," *International Mathematical Research Notices*, vol. 34, pp. 1853–1870, 2003.

[11] D. W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley, Sept. 1992.

[12] 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 (ICRA)*, May 2009.

[13] J. McLurkin and D. Yamins, "Dynamic Task Assignment in Robot Swarms," in *Proceedings of Robotics: Science and Systems*, Cambridge, USA, June 2005.

[14] L. Liu, B. Fine, D. Shell, and A. Klappenecker, "Approximate Characterization of Multi-Robot Swarm "Shapes" in Sublinear-Time," in *Proceedings of 2011 IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[15] S.-H. Cha, "Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions," *Inernational Journal of Mathematical Models and Methods in Applied Sciences*, 2007.