On Sensor Network Localization Exploiting Topological Constraints*

Alberto Speranzon[1], Shashank Shivkumar[1] and Robert Ghrist[2]

***Abstract*— We present a novel approach to localize an unknown planar sensor network based on sparse sampling of paths with minimal data. The problem is inspired by mapping the geometry of a building floorplan via "uncooperative sensing", using data from camera feeds and other tracking-capable sensors. Unique challenges include having no knowledge of sensor placement, coverage or their extrinsic parameters. The methods used are, at first, topological, to build a combinatorial model with the appropriate topology. This model is then augmented to include weak geometric information, and optimization techniques are used to approximate the domain.**

I. Introduction

The deployment of networked sensors in different environments, such as buildings, warehouses or manufacturing facilities, is improving both people's lifestyles, enabling remote monitoring and early detection of life threatening events (e.g. fires), as well as the safety and productivity of various industries.

Typical problems of interest in the context of sensor networks, range from estimation [1], [2], [3] to optimization [4], [5], and localization [6], [7], [8], to name a few.

Our paper is more strongly related to the localization problem, but in this paper we take a different perspective, in terms of: (1) type of data collected, (2) the localization process, and (3) the mathematical representation thereof.

In particular, we are concerned with the problem of building a map of an indoor environment using pre-existing sensors that we consider *uncooperative*, namely they are have been deployed in the environment for reasons that are not related to mapping. Many buildings possess such sensors, such as security cameras, proximity or motion sensors, readers, smoke detectors, smart locks, etc. By uncooperative sensors we mean that no information about their location or no knowledge of their parameters is available and that such sensors are sparsely distributed within the environment.

We assume that uncooperative agents are moving within the environment, triggering such sensors, which in turn generate observations/events that are available to the localization/mapping algorithm. In this paper we envision agents to be people who utilize the building and of course move with the goal of helping the map building process. One could potentially consider a mixed solution were simple, cooperative small and cheap robots, with minimal sensing capabilities, move in the environment triggering sensors, but are incapable of storing large quantities of data and thus of building maps.

It appears clear from the problem setup that this localization problem is different than the standard ones. In general, approaches either localize static sensors that directly measure relative distances or bearings with neighboring sensors or the sensors (or agents/robots) themselves move in the environment creating local maps that are then merged. This paper takes a different direction in the sense we consider static sensors that *do not directly* measure relative information. The approach we take is to first build a faithful *topological map* and then realize it in a two dimensional space.

Literature in robotics [9], [10], [11], [12] and sensor networks [13], [14], [15] tend to use "'topology" to refer to an underlying graph structure that represents either a map or the sensor network itself, respectively. As such, these representation are one dimensional, capable of only capturing the underlying connectivity of the space or network. In this paper we will instead build higher-dimensional representation using *cell complexes*. This construction, while certainly more computationally demanding, ultimately enables us to deal with the uncooperative nature of the problem and realize a 2D map despite the sparsity of sensing.

A. Related Work

The topic discussed in this paper is related to mapping problems in robotics, localization of sensor networks and work on camera networks. We briefly review the key differences between our work and the literature.

The problem of generating a map—while localizing the agent in it—is a well established area in robotics called SLAM (Simultaneous Localization and Mapping). A recent survey [16], provides a comprehensive overview of the methods developed in this area. Among them, topological mapping is one possible sparse representation of the environment. Compared to the definition in [16], where topology and geometry are "mixed", the representation we consider in this paper is strictly topological first—a cell complex—and only later, by adding metric/geometric information, we realize it a (two) dimensional space, ultimately an (embedded topological) map, as meant in [16] and references therein.

Compared to recent methods in robotics, such as [11], [17], where topological maps are built from dense information collected by cameras or LIDARs, through a "sparsification" process, with the purpose of reducing the otherwise overwhelming amount of data, the approach we consider

*This work was supported with funding from DARPA and the AFRL under agreement number FA8650-18-2-7840. The views, opinions and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

[1]A. Speranzon and S. Shivkumar are with Honeywell Aerospace, Advanced Technology, Plymouth, MN 55441, USA. firstname.lastname@honeywell.com.

[2]R. Ghrist is with the Department of Mathematics, University of Pennsylvania, Philadelphia, PA 19104, USA. ghrist@math.upenn.edu.

here leverages data streams that are collected by a sparse set of sensors, resulting in information that it sparse both temporally and spatially.

Similar to the approach considered here, in [18] the authors represent maps as (simplicial) complexes. The focus of this work is mostly on the development of a multi-agent protocol that enables a swarm of simple robots to build a topological map, without GPS and with minimalistic sensing. Another paper, similar in spirit, is [19], where the authors combine ideas from persistent homology and computational geometry (Voronoi partitions) to derive a topological map.

Related to the problem of sensor network localization, see [6], [20], [7], [8] and references therein, this paper leverages a similar optimization framework, but it augments it with constraints obtained from the topological construction. In particular, lower-bounds are added to the optimization problem obtained from non-trivial cycles of the cell complex. The benefits is that we obtain better realization despite the sparsity of information.

Research in the area of camera networks is also related to our work. Typical problems are mostly related to tracking people across the network, synchronization, etc. Related to our work, given that it leverages similar models, namely simplicial complexes (a restriction of cell complexes used in this paper), is the work in [21]. However, the problem there is about coverage estimation rather than mapping. In [22] the authors consider the problem of estimating the poses of camera sensor network based overlapping views and with known intrinsic parameters. We deal, in this paper, with a setting where camera views are not overlapping and where camera parameters are not known.

Overall, this paper distinguishes itself from this body of literature, in several ways. First of all, the sensors are not carried by the agents but instead the sensors (cameras) are "excited" by agents moving in a complex environment, thus the *agents are the observations*. Secondly, we construct a general cell complex representation that enables us to discover information about "non-traversable" areas of the building. Of course, as the complexity of the environment grows and the density of observations remain constant, it becomes more difficult to obtain faithful maps. Thirdly, we leverage topological information (non-trivial cycles) to pose an augmented graph realization problem, thus yielding a better geometric approximation of the environment, despite the underlying graph not being uniquely realizable. Finally, we provide mathematical conditions under which the cell complex faithfully captures the underlying structure of the building.

The paper is organized as follow. In Section II we formulate the overall problem and summarize key assumptions. Section III introduces the cell complex, namely a topological representation of the environment, which is then, as described in Section IV realized, via a semidefinite program, into a two dimensional space, once geometric information is added. Numerical results are summarized in Section V and concluding remarks are provided in Section VI.

II. Uncooperative Mapping Problem

Consider the problem of estimating a 2D map of a building using data streams from sensors present within the building, knowing neither their locations nor parameters. Potential examples include cameras, ID sensors, etc.

As people move within a building, they "excite" the sensors. Paths which transit from one sensor to another over time reveal underlying topological properties of the environment. Mathematically the problem can be formulated as follows: given a sparse sampling of the path space PX of an unknown planar domain X, with some additional data on when those paths happen to be closed in X, to what extent can X be inferred?

Assumption(s). *We make the following assumptions:*

A1: *Agents*[1] *detected by a sensor are uniquely identified;*
A2: *Sensor data feeds uniquely identified;*
A3: *Sensors have synchronized clocks: time is global.*

Independently of the specific sensor type, we have the following definition.

Definition II.1 (Events)**.** *An* event *is a triple* $(t, \mathrm{id}, \mathrm{sensor}) \in \mathbb{R}_+ \times \mathcal{P} \times \mathcal{S}$*, where* t *is the time when an event was recorded at a sensor,* id *is the unique identifier associated to a given person, who triggered the sensor, and* sensor *is the unique identifier of the sensor that captured the event. Here* $\mathcal{P} \subset \mathbb{N}$ *is a set of unique identifiers, one for each person, and* $\mathcal{S} \subset \mathbb{N}$ *is the set of unique sensor identification numbers. We define the space of events to be* $\mathcal{E} \subset \mathbb{R}_+ \times \mathcal{P} \times \mathcal{S}$*.*

In what follows, we focus on cameras, although much of the method is adaptable to other sensors. Generally, the boundary of a camera field of view (FOV) might be clustered into disconnected components. In Figure 1 we represent the FOV of each camera with a shaded region. At it can be seen the FOV of each camera is "broken" into two (green/orange FOV) and three (blue FOV) component boundary. Locally, each camera can assign a unique identifier to each of its (possibly disconnected) boundaries. We will then indicate the boundaries of the FOV of a sensor i by $\{C_{i,1}, \ldots, C_{i,n}\} \in \mathcal{S}$. Figure 1 shows three cameras, C_1, C_2 and C_3. The FOV is depicted here (for simplicity) as a circle sector. People, indicated as p_i, $i \in \{1, 2, 3\}$, move within the building entering and exiting the FOV of the cameras, generating unique events $(t, p_i, C_{j,\ell})$.

Assumption(s). *We further assume that:*

A4: *every path followed by a person is continuous in the domain* X*.*

Remark II.1. *Although this assumption appears to be superfluous, as it seems to be always verified, it can actually be violated if, for example, a person is able to use an elevator to a different floor which is not monitored, and then by taking another elevator she/he appears in a different part of the*

[1] Including the case of multiple agents triggering the same sensor at the same time – not unrealistic for optical sensors given recent advances in Convolutional Neural Networks [23].

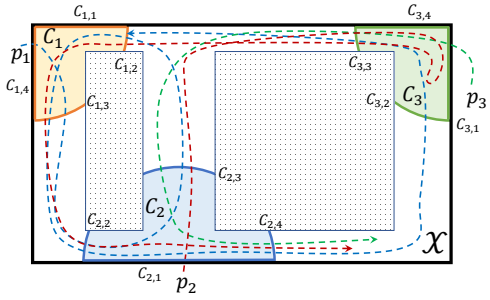

Fig. 1. Cameras are indicated by C_i and their field of view depicted with an idealized circular sector. Boundaries of the FOV of camera C_i are denoted $C_{i,\ell}$. Trajectories followed by three different uniquely identifiable people are indicate with p_1, p_2 and p_3. Note that certain shaded areas are not traversable. These obstacles induce topological features in the domain.

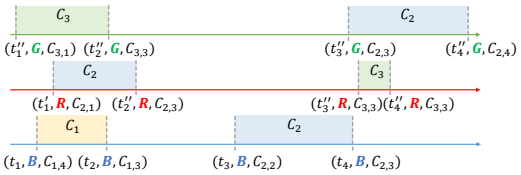

Fig. 2. (a) Observations in the spatial domain: note that these are embedded in the Euclidean 2D space assuming we knew the coordinates of the events, which however are not known and we wish to reconstruct. (b) Corresponding events (first four for each person) in the time domain: these are the actual data we have available. We show with a colored rectangle the times when a person is in the FOV of the corresponding camera.

starting floor. This would clearly appear as a discontinuity in the path of a person walking in the starting floor.

Under the assumptions (A1)–(A4) we can define consecutive events:

Definition II.2 (Consecutive Events)**.** *Given continuous paths over the domain $\mathcal{X}$, two events $(t, p_i, C_{j,r})$ and $(t'', p_i, C_{k,s})$ with $t < t''$ are said to be consecutive. We will write $C_{k,s} \succ C_{j,r}$.*

Figure 2 shows the type of observations that are available (where only the first four events for each person are listed) for the mapping. Comparing these with Figure 1 we note that the observations are very sparse in the spatial domain, compared to the underlying continuous process (paths), as these are only recorded at the boundary of the FOV of a camera. Of course we also can keep track of people within the FOV.

From Figure 2 we see that $C_{2,3} \succ_G C_{3,3}$ and $C_{3,3} \succ_R C_{2,3}$ (top two plots) and $C_{2,2} \succ_B C_{1,3}$ (bottom plot), where R, G, B stands for *Red*, *Green* and *Blue*. Given a set of events $\{(t_k, p_i, C_{j,\ell})\}$ we wish to infer the map. We consider the following definition of an $\mathbb{R}^2$*-realized map*.

Definition II.3 ($\mathbb{R}^2$-realized map)**.** *An $\mathbb{R}^2$-realized map is a framework* [2] *$(\mathcal{G}, p) = ((V, E), p)$ where $V = \mathcal{P}$ is a set of vertexes corresponding to distinct FOV boundaries, $E \subset \mathcal{S} \times \mathcal{S}$ is a set of edges, where an edge connects two boundaries if they are associated to consecutive events and $p : V \to \mathbb{R}^2$ is a map that assigns coordinates to vertexes such that $\|p(v_i) - p(v_j)\|_\ell = \hat{d}_{ij}$ where $\hat{d}_{ij}$ is an estimate of the ℓ-norm between the points.* [3]

Remark II.2. *Given that we have no direct observability of agent paths, but only an ordered set of events, it is unrealistic to expect an accurate map of the unobservable space. Even when the topological map is faithful—we will characterized conditions when this is not possible—there is an inherent error in estimating the distance from a camera to another one based exclusively on the time interval between consecutive events.*

Note that we are also realizing a graph whose vertices are the boundaries of the FOV. Indeed, embedding the boundaries of the FOV of comeras, enables us to estimate a better map given that to each camera we can, in general, associate multiple boundaries.

III. Topological Maps & Cell Complexes

For notational simplicity, we will rename all FOV boundary components $C_{i,r}$ with a unique single index. Given a set of events $\{(t_k, p_i, C_j)\}$, we can estimate the transition time between two consecutive events as $T^i_{jk} = \{\{t_k - t_j, p_i, C_{jk}\} | C_k \succ_i C_j\}$. We can define non-negative real-valued random variables $t^i_{jk} : \{T^i_{jk}\} \to \mathbb{R}_{\geq 0}$ as being a Borel measurable function defined on the event set $\{T^i_{jk}\}$. This means that to each transition C_{jk} we can associate a random variable $t^i_{jk} \sim f_{t_{jk}|p_i, C_{jk}}(t | p_i = i, C_{jk} = \{C_k \succ_i C_j\})$, where $f_{t_{jk}|p_i, C_{jk}}$ is the conditional probability distribution of transit times given that the transition is from camera C_j to camera C_k and the person transitioning is i.

Given the transit time t^i_{jk}, the pdf associated to it can either be unimodal or multi-modal. In this last case, we decompose it into a mixture of unimodal pdfs. Multi-modality can be detected using standard statistical methods, such as Hartigan's dip statistics [25] or similar methods, and the decomposition is done using a mixture model: see [26] for a recent discussion. We thus have that the pdf of the random variables t^i_{jk} is decomposed by a set of pdfs $\{f_\ell(t | p_i = i_\ell, C_{jk})\}$, such that $f(t | p_i = i, C_{jk}) = \sum_\ell w_\ell f_\ell$ where $w_\ell > 0$ and $\sum w_\ell = 1$. We therefore represent the transition time t^i_{jk} with a set of random variables $\{t^{i_\ell}_{jk}\}$ each with unimodal distributions.

We start by associating to the observations a skeletal map described as a *multi-graph* or, equivalently, a 1-dimension *cell complex*.

A multi-graph is a combinatorial graph which allows for multiple edges between the same pair of vertices (or between the same vertex). More formally, a (directed) multi-graph is a tuple $\mathcal{G} = (V, E, \mathrm{src}, \mathrm{tgt})$ where, V is the set of vertices, E is the set of edges, $\mathrm{src}, \mathrm{tgt} : E \to V$ are the *source* and *target* functions. Namely for an edge $e \in E$, $\mathrm{src}(e) = v_1$ and $\mathrm{tgt}(e) = v_2$ means that the edge e starts at vertex v_1 and ends at v_2.

We are now in the position to define a *multi-graph representation* of our sensor data.

[2] We use here a terminology that is common in graph rigidity theory [24].

[3] For an office building the $\ell = 1$ norm seems especially apt.

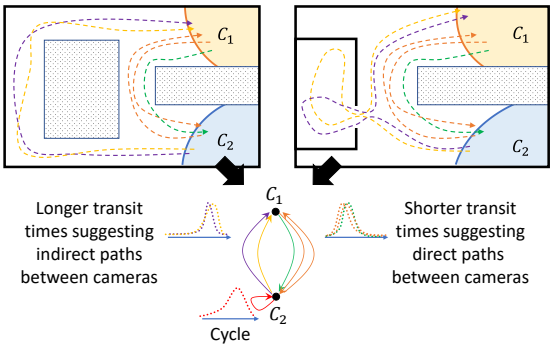

Fig. 3. Two scenarios that lead to different transit time distributions. Note that although the situation on the left might appear unrealistic in practice, this could actually occur if vending machines or a water fountain is present on the leftmost corridor.

Definition III.1 (Multi-Graph Representation)**.** *A* multi-graph representation *of the domain X, is a* weighted *(directed) multi-graph $\mathcal{M} = (V, E, \mathrm{src}, \mathrm{tgt}, \mathcal{T})$ where $V = \{C_j\}$, $\mathcal{T}(e_{jk}) = \{t_{jk}^{i_\ell}\}$ are unimodal random variables and $e_{jk} \in E$ and such that:*

$$\mathrm{src}(e_{jk}) = C_j\,, \mathrm{tgt}(e_{jk}) = C_k \iff C_k \succ C_j\,.$$

Note that $\mathcal{M}$ captures the connectivity of the domain $\mathcal{X}$. Note, however that, compared to classical topological methods, such as in robotics [10], such a representation is a (weighted directed) multi-graph. The main reason why we take into consideration the transit time distributions from one FOV to another is two-fold: on one side it provides a proxy for geometric distance, which will be used later on to geometrically realize $\mathcal{M}$; on the other, it enables us to "cluster" transit times and collapse some of the edges.

The construction of $\mathcal{M}$ is far from parsimonious. Our next step is to cluster unimodal distributions across people to extract robust and relevant topological information about the unobservable part of the domain $\mathcal{X}$.

The reader might, at this point, wonder why we have considered a decomposition of the transit times and then a clustering. To clarify, let us consider the scenarios in Figure 3. The presence of a separate corridor (left) or of a room (right) with no sensor coverage would be undetectable if we were not to decompose and cluster unimodal transit time distributions.

Under the assumption that one collects data over long periods of time (e.g. months/years) we expect distributions to have very "sharp peaks" around a few critical times, as depicted in Figure 3.

It is clear that the multi-graph $\mathcal{M}$ likely contains redundant information as some edges have "similar" transit times distributions, see the multi-graph in Figure 3. These can be combined into a single edge. Formally we can iteratively apply a *collapsing map* to $\mathcal{M}$ to obtain a *reduced multi-graph* $\mathcal{M}_R = (V_R, E_R, \mathrm{src}_R, \mathrm{tgt}_R, \mathcal{T}_R)$.

It is convenient at this stage also make $\mathcal{M}$ undirected so that if $\mathrm{src}(e) = i$ and $\mathrm{tgt}(e) = j$ then there exists $e' \in E$ such that $\mathrm{src}(e') = j$ and $\mathrm{tgt}(e') = j$. We will assign to e' the same transit distribution as to e. This step is well motivated given that we are constructing a map of the environment. So, while there might be a preferential direction of motion, geometrically these points are clearly connected.

Consider the collapsing map $\rho^\alpha : \mathcal{M}^\alpha \to \mathcal{M}^{\alpha+1}$, for step $\alpha = \{0, \dots, n-1\}$ where $\mathcal{M}^0 = \mathcal{M}$ and $\mathcal{M}^{n-1} = \mathcal{M}_R$. The map $\rho^\alpha = \langle \rho_V^\alpha, \rho_E^\alpha, \rho_{\mathcal{T}}^\alpha \rangle$ is such that $\rho_V^\alpha = \mathbb{1}$ (identity map) for all α. For edges and transit time we have $\rho_E^\alpha : E^\alpha \times E^\alpha \to E^{\alpha+1}$ and $\rho_{\mathcal{T}}^\alpha : \mathcal{T}^\alpha \times \mathcal{T}^\alpha \to \mathcal{T}^{\alpha+1}$, such that, for any $e, e' \in E^\alpha$, $e \neq e'$ with $\mathrm{src}(e) = \mathrm{src}(e')$ and $\mathrm{tgt}(e) = \mathrm{tgt}(e')$ and corresponding transit times t_e and t'_e, then:

$$(\rho_E^\alpha(e, e'), \rho_{\mathcal{T}}^\alpha(e, e')) = \begin{cases} (e'', t_{e''})\,, & \text{iff } \mathrm{dist}(t_e, t_{e'}) \leq \delta\,, \\ (e, t_e)\,, & \text{otherwise}\,. \end{cases}$$

where e'' is a new edge between the same vertices as e and e' ($\mathrm{src}(e) = \mathrm{src}(e') = \mathrm{src}(e'')$ and $\mathrm{tgt}(e) = \mathrm{tgt}(e') = \mathrm{tgt}(e'')$). The function $\mathrm{dist}(t_e, t_{e'})$ is a suitable distance metric (e.g. the Wasserstein distance [27]) between the transit time pdfs t_e and $t_{e'}$ associated to e and e' respectively. We have that the pdf of $t_{e''}$ is given as $f_{t_{e''}} = \gamma^{-1} f_{t_e} \cdot f_{t_{e'}}$, as we assume t_e and t'_e independent and γ a normalizing factor.

Given that we deal with finite multi-graphs, we have the following straightforward result: composing the collapsing maps ρ^α yields a well-defined reduced multi-graph representation[4]. Furthermore, there is an upper bound on the number of steps required to generate it.

Proposition III.2. *The collapsing maps $\rho^\alpha : \mathcal{M}^\alpha \to \mathcal{M}^{\alpha+1}$, for $\alpha = \{0, \dots, n-1\}$, with $\mathcal{M}^0 = \mathcal{M} = (V, E, \mathrm{src}, \mathrm{tgt}, \mathcal{T})$ and $\mathcal{M}^{n-1} = \mathcal{M}_R$ terminates with $n \leq \max_{v_1, v_2 \in V} |\{e \in E : \mathrm{src}(e) = v_1, \mathrm{tgt}(e) = v_2\}|$.*

The multi-graph $\mathcal{M}_R$, while more compact, is still a one-dimensional topological representation of $\mathcal{X}$ which captures the connectivity of the space. However, there is higher-dimensional information we can leverage. Motivated by the fact that a multi-graph is in fact a 1-dimensional version of a *cell complex*, we pass from a combinatorial description to a more explicitly topological structure.

Cell complexes are topological spaces assembled from parts, much like a multi-graph is assembled from vertices and edges. These come in several varieties, depending on the types of cells used and the constraints on how they are assembled [28], [29]. A cell complex is built inductively as follows

- Start with the 0-skeleton: a discrete set of 0-cells (vertices), $X^{(0)}$.
- The k-skeleton $X^{(k)}$ is obtained from $X^{(k-1)}$ by attaching closed k-cells, e_α^k, via *attaching maps* $\phi_\alpha : \partial e_\alpha^k \to X^{(k-1)}$. Namely, $X^{(k)} = X^{(k-1)} \bigsqcup_\alpha e_\alpha^k / \sim$, where $\sim$ is the equivalence relation $x \sim \phi_\alpha(x)$ for $x \in \partial e_\alpha^k$. This means that we are identifying as equivalent, boundaries of cells with images in the lower dimensional skeleta.
- The completed cell complex X is the union of its skeleta: $X = \cup_k X^{(k)}$. We exclusively use finite complexes.

[4]Suitable choices of the kernels for multi-modal decomposition and of a distance between pdfs is, in general, required to ensure the commutativity of the composition of ρ^α.

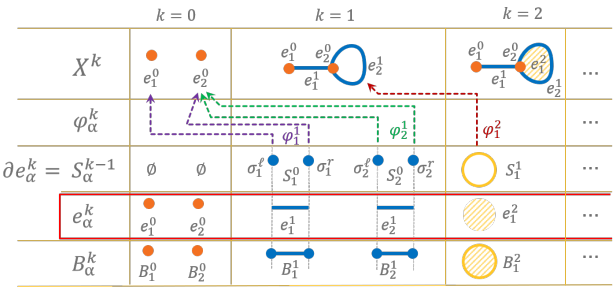

Fig. 4. Example of the construction of a cell complex. We indicated with B^k_α the closed cell of order k. Note that the attaching maps allow us to "glue" an (open) edge to a single node to form a cycle or to glue an (open) disc to a loop, closing the loop.

Figure 4 shows a few steps of the construction. In the case of e^k_α a closed k-simplex and each ϕ_α a face-preserving homeomorphism, one has the familiar notion of a (geometric) *simplicial complex*; relaxing certain of these assumptions leads to a Δ-*complex*; the most general useful case, in which e^k_α is a closed k-dimensional ball and ϕ_α is merely continuous, leads to a *CW complex*. Each has its advantages and disadvantages [28].

It is this structure we use to interpolate edges in the multi-graph representation with higher-dimensional cells that "fill in" holes, in order to approximate the domain topology. Our construction is reminiscent of an old idea from Dowker [30].

1) Vertices (0-cells) correspond to the set of components of FOV boundaries $\{C_i\}$, representing these as abstract points. Edges and higher-dimensional cells will typically be of the form of a closed cell Δ^k based on $k+1$ vertices which are related by the transit data in a particular manner.
2) With sufficient crossings of agents in each camera FOV, one expects a complete connected subgraph on the vertices within that camera's FOV. Since a camera's FOV is a star-convex region, fill in this complete connected subgraph with a single closed simplex whose 1-skeleton is that subgraph.
3) For all remaining edges in the reduced multi-graph $\mathcal{M}_R$ that are *exterior* to the camera FOVs we proceed as follows: for each (unordered) pair of (not necessarily distinct) vertices $\{v_1, v_2\}$, order the set of edges $\{e_{1,2}\}$ according to statistics on the transit time, e.g. we can consider the mean of the transit time distributions. Thus we have that these edges can be uniquely *indexed* by $(1, 2, 3, \ldots)$.
4) Close off all self-loops (staring and ending at the same FOV), in $\mathcal{M}_R$, by attaching a 2-disc. Remove these edges from further consideration.
5) Beginning with all remaining index-1 exterior edges, find all complete connected subgraphs and complete each to a closed simplicial cell as done in the internal FOV above.
6) From this set of edges, increment each to the next highest index; or, if there is none, keep it under consideration. Repeat the process of filling in maximal cells and ratcheting to "longer" edges until the set is exhausted.

Proposition III.3. *The above construction produces a cell complex whose homotopy type is that of a graph. In the restricted setting of all distributions unimodal and $\mathcal{X}$ a planar domain with unobserved region $\mathcal{X} - \cup_i C_i$ being a finite disjoint union of simply-connected pieces, then the above construction yields a cell complex homotopy-equivalent to $\mathcal{X}$.*

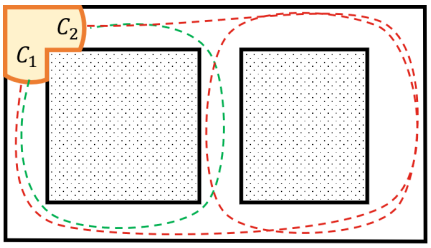

Fig. 5. Example where the unobserved region is not a simply connected space. In this case paths that encircle the rightmost non-traversable region would generate highly multi-modal distributions leading each mode (corresponding to a loop) leading to the belief that there is a distinct non-traversable region.

The interpretation of this result is that under reasonable assumptions, the cell complex will have non-trivial cycles corresponding to non-traversable regions of the domain (dotted regions Figure 1.

It is neither asserted nor true that the topology of the cell complex matches the topology of $\mathcal{X}$ in general, since we have no assumptions on sensor coverage: in extreme cases, there are "too many holes" in the unobserved regions. A situation that violates the previous proposition is shows in Figure 5. In this case the "loopy" paths on the right, would be assigned multiple non-trivial cycles.

Nevertheless, with our construction, a surprisingly small number of cameras suffices to accurately reconstruct domain topology in many cases.

IV. Geometric Information: Map Realization

As mentioned, the cell complex provides a topological representation of the environment. To add geometric data, we need to be able to assign to each vertex a coordinate in $\mathbb{R}^2$. Such coordinates will need to ensure that the information captured over edges is consistent with the weight—transit time—over that particular 1-cell. Assuming a typical motion velocity for agents within a building (which can also be estimated by observing people moving within the FOV of the cameras), we can approximate distances from transient times. This evokes the *graph realization* problem [31]. Although NP-hard [32], it can be (approximately) solved efficiently and in some special cases optimally.

We discuss next how we capture information about the cell complex within a graph realization problem yielding an embedding of the cell complex representing the map.

One advantage we have lies in higher dimensional information—2-cells—which we can leverage to "strengthen" the two-dimensional embedding. In particular, we consider minimal non-trivial cycles of the 2-skeleton of the cell complex. These are the sets of cycles that encircles a non-contractable space (practically, non-traversable regions of a building).

As it is not possible to realize a multi-graph (we can have multiple distanced between two vertices) we will consider a barycentric subdivision of it [28]. Namely, for 1-cells of index-i, $i > 1$, we add one (or more) vertices on the 1-cell. Thus the resulting structure is simple graph and the embedding problem then becomes very similar to a graph realization, where, given a set of relative distances between a set of vertices, connected by edges, we are interested to find coordinates that satisfy such distance constraints[5].

Compared to the classical graph realization problem, however, we consider cycles in the problem. These are modeled, in the optimization problem, as a set of lower bounds on the distance between two diagonally opposite vertices of a cycle.

If we let $P = (p_1, \dots, p_m)^T \in \mathbb{R}^{m\times 2}$ be the coordinates of the vertices, then we can construct the rank-2 matrix $Y = PP^T$, and we can pose the following SDP [20]:

$$\begin{aligned} \text{maximize} \quad & 0 && (1)\\ \text{s.t.} \quad & Z_{1:d,1:d} = I_d \\ & e_{ij}e_{ij}^T \bullet Z = d_{ij}^2\,, \;\; a_{kh}a_{kh}^T \bullet Z = d_{kh}^2 \\ & h_{rs}h_{rs}^T \bullet Z \geq \bar{d}_{rs}^2 && (2)\\ & Z \succeq 0 \end{aligned}$$

Here we denote $\text{trace}(A^TB) = A \bullet B$. In this optimization problem, we have denoted $e_{ij} = (\mathbf{0}, e_i - e_j)^T$ with e_ℓ being the vector of all zeros except the ℓ-th element, which is one. In addition, $a_{kh} = (-f_k, e_h)^T$ where f_k are the coordinates of fixed vertices (*anchors*). We have that $Z = YY^T$ and the rank constraint has been relaxed as a semidefinite constraint.

The main difference here, compared to standard graph realization problems is the presence of the constraints (2). We explain next the meaning and rationale for these.

It is well known [33], that the problem (1) will have a unique solution if the underlying graph is *generally globally rigid*. Roughly speaking, this means that there are no vertices or subgraphs that can be translated, rotated or reflected with respect to other vertices/edges, without violating distance constraints. The presence of non-traversable regions, however, leads to issues if we consider the standard graph realization problem formulation [31], [20], because subgraphs enclosing such regions are not uniquely localizable. A way to think about this, is that there are too few edges in such subgraphs and local motions of (some of) the vertices yield many realizations that satisfy the distance constraints.

A way to improve this, is by leveraging the cell complex construction and the fact that we know which cycles "enclose" non-traversable regions. We can then associate to such cycles inequality constraints capturing the fact that opposite vertices belonging to cycles should be "far apart".

The constraints (2), represent such a requirement where we have:

$$h_{rs} = \begin{cases} (\mathbf{0}, e_{rs})^T, & \text{if neither } r \text{ or } s \text{ are anchors}\,,\\ (-f_r, e_s)^T & \text{if } r \text{ is an anchor and } s \text{ is not}\\ & \text{(swapping indices if } s \text{ is an}\\ & \text{anchor and } r \text{ is not)} \end{cases}$$

and where $\bar{d}_{rs}$ is a user-defined value. Of course, in general this is not known. However, a guess, based on the sides of the cycle, can be chosen.

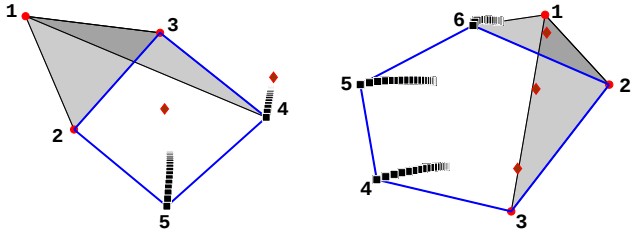

Fig. 6. Two simple examples showing the benefit of the approach to realize graphs that are not globally rigid. In blue we show non-trivial cycles of the cell complex. 2-cells are shown as shaded gray areas. In black we show the realization of the graph when we use the constraints (2), with $\bar{d}_{rs} = \epsilon d_{rs}$, where $\epsilon \in [0.1, 1]$, with d_{rs} being the true distance. Red diamonds show the realization when no inequality constraints are used.

V. Numerical Results

Before we show how the proposed method works in a building-like setting, we show the advantage of constructing a higher dimensional cell complex to improve the graph realization.

We consider two toy examples, as shown in Figure 6. Here we show two complexes where we have the nodes 1, 2, 3 being anchors, two 2-cells indicated with the shaded gray regions and in blue edges (1-cells). The known distances are the edges in black and blue. If we were to solve a graph realization problem with only this information the result is shown with the red diamonds. We used SeDuMi [34] SDP solver, with Yalmip [35] as interface, to obtain the results. Since the overall graph is not uniquely realizable we get an arbitrary realization, far from the true locations. If we consider the cycles (shown in blue) and for each cycle we add the constraints (2) (lower-bound on the diagonals of a cycle), we obtain the realizations shown with black squares. Here we show 20 realizations for the lower bound being 10% to 100% the true distance. Of course for when we know exactly the true distance of vertices on the diagonals, the graph is globally rigid and (1) yields a perfect realization.

In Figure 7 we show the results for more complicated scenarios. In this case we have multiple cameras, denoted with colored sectors. The gray dotted lines show the distance measurements between the camera FOVs. In these simulations we have considered the noiseless case and postpone results with noise to future work[6].

[5]We consider here the Euclidean distances obtained from the the mean of the transit time distribution, times the nominal walking speed of a person. We leave the discussion of other norms (potentially more suitable but also more complicated to deal with, because of non-uniqueness, such as ℓ_1) to future extensions.

[6]The setting with noise can be addressed by extending the literature on graph realization with noisy distances [36] in a similar manner as discussed here, where we introduce extra constraints associated to cycles. While the extension appears to carry over in a fairly straightforward manner, we postpone a detailed discussion as future work.

At the top of Figure 7, we consider a scenario with four cameras and thee non-traversable areas. We indicate with gray dotted lines the (noiseless) measurements between different camera FOVs. Cycles are indicated with blue lines and can be computed by constructing the cell complex described in Section III. The cycles are (2, 3, 8, 9), (5, 6, 10, 9) and (2, 5, 12, 11). All the other cycles are "filled in" by higher order cells (we do not depict those otherwise the picture would be too crowded). We assume that vertices 1, 4, 13 (entries to the buildings) have a known location (anchors). The solutions of the graph realization problem, with and without the inequality constraints (2) are shown with diamond and square, respectively. As it can be seen, while there is some scale mismatch, the squares are closer to the truth. In this simulation we have considered that the lower bounds are 70% of the true distant plus/minus 5% (thus each $\bar{d}_{rs}$ is different). This simulates a situation where we need to assume some value for the $\bar{d}_{rs}$ and we use imprecise values. Of course, larger mismatches will lead to realizations in between the red diamonds and black squares. The improvement in this case, considering the RMSE, is about 24% better than standard graph realization methods.

At the bottom Figure 7, we consider the scenario where there is are two paths between 2 and 5. Note that there are no cameras in the top part of the building and thus transit times between 2 and 5 will have two different statistical distributions. In the case we do not have paths looping around the top non-traversable region, the cell complex construction does recover the presence of an extra loop (2, 5, 12, 11). Note that here we have considered a barycentric subdivision with two extra virtual nodes enabling us to capture more constraints. Of course this does require to know lower bounds on the distance between 5, 11 and 2, 12. If this was not available one could have split the long 2-5 edge into two parts. In this case, although the realization considering the inequalities coming from the cycles, improve the results, nodes 11 and 12 are embedded at the center of the building, because of the reflection uncertainty: inequalities and equality constraints are all satisfied independently if 11 and 12 are above or below the edge 2-5. Overall, the results considering cycles are again better that when no cycles are used. In this case the improvement is lower, with the RMSE that is about 18% better than standard graph realization methods.

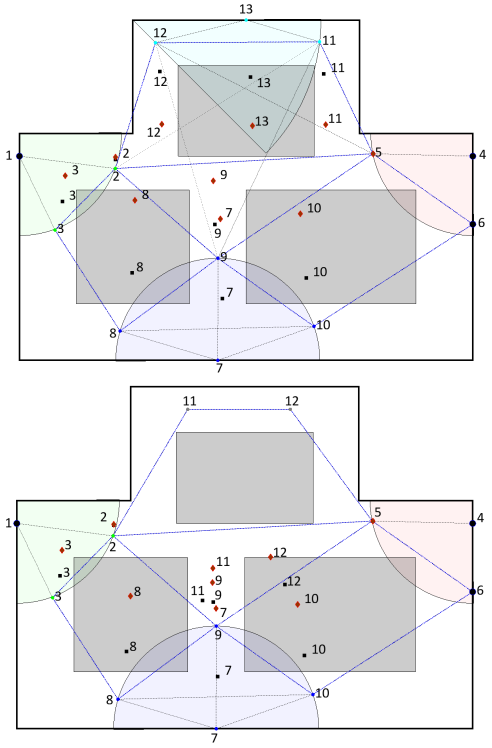

Fig. 7. With dark orange diamonds (♦) we show the result when we solve a SDP problem without considering lower bounds (2), provided by the cycles. Cycles encircle non-traversable regions and are shown with dotted blue lines. With dark square (■) we show the proposed approach when we do consider lower-bounds. Here we assume that the lower bound is fixed to be $70\% \pm 5\%$ of the true distance. We assumed that some of the entry points have known location (anchors) and we have indicated them with large black dots (●). Note that at the bottom of the second figure, the points 11 and 12 are embedded towards the center of the building. This is because there is a reflection uncertainty of vertices 11 and 12 with respect to the edge 2-5.

VI. Conclusions

This paper demonstrated a novel method for topological mapping in the context of minimal sensing. The test case investigated involved surreptitious control of security camera feeds combined with agent-tracking in order to reconstruct domain topology and geometry through inference of minimal paths. Inference of domain topology occurs through construction of an appropriate cell complex, whereas geometric approximation was done through solving a semidefinite program. Neither method has guaranteed accuracy, given the possibility of too little sensor data as input. The geometric approximation suffers from potential existence of local optima for the SDP that do not reflect ground truth geometry, however we demonstrate that the addition of constraints in the SDP yields better results than using standard methods. Future work includes the testing on more complex and realistic simulated data and the extension to multi-floor buildings.

References

[1] W.-A. Zhang and L. Shi, "Sequential fusion estimation for clustered sensor networks," *Automatica*, vol. 89, pp. 358–363, 2018.

[2] G. Battistelli and L. Chisci, "Stability of consensus extended kalman filter for distributed state estimation," *Automatica*, vol. 68, pp. 169–178, 2016.

[3] A. Speranzon, C. Fischione, K. H. Johansson, and A. Sangiovanni-Vincentelli, "A distributed minimum variance estimator for sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 609–621, 2008.

[4] A. Nedic, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM Journal on Optimization*, vol. 27, no. 4, pp. 2597–2633, 2017.

[5] P. Di Lorenzo and G. Scutari, "Next: In-network nonconvex optimization," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 2, pp. 120–136, 2016.

[6] B. Jiang, B. D. O. Anderson, and H. Hatem, "3D relative localization of mobile systems using distance-only measurements via semidefinite optimization," *IEEE Transactions on Aerospace and Electronic Systems*, 2019, to Appear.

[7] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements," in *Proceedings*

of the 2nd international conference on Embedded networked sensor systems. ACM, 2004, pp. 50–61.

[8] L. Doherty, L. El Ghaoui, *et al.*, "Convex position estimation in wireless sensor networks," in *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No. 01CH37213)*, vol. 3. IEEE, 2001, pp. 1655–1663.

[9] B. Kuipers and Y.-T. Byun, "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations," *Robotics and autonomous systems*, vol. 8, no. 1-2, pp. 47–63, 1991.

[10] S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation," *Artificial Intelligence*, vol. 99, no. 1, pp. 21–71, 1998.

[11] F. Blöchliger, M. Fehr, M. Dymczyk, T. Schneider, and R. Siegwart, "Topomap: Topological mapping and navigation based on visual slam maps," *arXiv preprint arXiv:1709.05533*, 2017.

[12] H. Oleynikova, Z. Taylor, R. Siegwart, and J. Nieto, "Sparse 3d topological graphs for micro-aerial vehicle planning," *arXiv preprint arXiv:1803.04345*, 2018.

[13] B. Deb, S. Bhatnagar, and B. Nath, "A topology discovery algorithm for sensor networks with applications to network management," 2002.

[14] R. Olfati-Saber and J. S. Shamma, "Consensus filters for sensor networks and distributed sensor fusion," in *Proceedings of the 44th IEEE Conference on Decision and Control*. IEEE, 2005, pp. 6698–6703.

[15] A. Nedić, A. Olshevsky, and M. G. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 953–976, 2018.

[16] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[17] P. Schmuck, S. A. Scherer, and A. Zell, "Hybrid metric-topological 3d occupancy grid maps for large-scale mapping," *IFAC-PapersOnLine*, vol. 49, no. 15, pp. 230–235, 2016.

[18] R. Ramaithitima, M. Whitzer, S. Bhattacharya, and V. Kumar, "Automated creation of topological maps in unknown environments using a swarm of resource-constrained robots," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 746–753, 2016.

[19] R. K. Ramachandran, S. Wilson, and S. Berman, "A probabilistic approach to automated construction of topological maps using a stochastic robotic swarm," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 616–623, 2017.

[20] A. M.-C. So and Y. Ye, "Theory of semidefinite programming for sensor network localization," *Mathematical Programming*, vol. 109, no. 2-3, pp. 367–384, 2007.

[21] E. Lobaton, R. Vasudevan, R. Bajcsy, and S. S. Sastry, "A distributed topological camera network representation for tracking applications," *IEEE Transactions on Image processing*, vol. 19, no. 10, pp. 2516–2529, 2010.

[22] R. Tron and R. Vidal, "Distributed image-based 3-D localization of camera sensor networks," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*. IEEE, 2009, pp. 901–908.

[23] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[24] R. Connelly, "Generic global rigidity," *Discrete & Computational Geometry*, vol. 33, no. 4, pp. 549–563, 2005.

[25] J. A. Hartigan and P. M. Hartigan, "The dip test of unimodality," *The annals of Statistics*, vol. 13, no. 1, pp. 70–84, 1985.

[26] J. W. Miller and M. T. Harrison, "Mixture models with a prior on the number of components," *Journal of the American Statistical Association*, vol. 113, no. 521, pp. 340–356, 2018.

[27] A. L. Gibbs and F. E. Su, "On choosing and bounding probability metrics," *International statistical review*, vol. 70, no. 3, pp. 419–435, 2002.

[28] A. Hatcher, *Algebraic topology*, 17th ed. Cambridge University Press, 2001, Available online: (web).

[29] R. Ghrist, *Elementary applied topology*. Createspace Seattle, 2014, vol. 1, Available online: (web).

[30] C. H. Dowker, "Homology groups of relations," *Annals of mathematics*, pp. 84–95, 1952.

[31] A. M.-C. So, "A semidefinite programming approach to the graph realization problem: Theory, applications and extensions," Ph.D. dissertation, Stanford University, 2007.

[32] J. B. Saxe, *Embeddability of weighted graphs in k-space is strongly NP-hard*. Carnegie-Mellon University, Department of Computer Science, 1980.

[33] B. Jackson and T. Jordán, "Graph theoretic techniques in the analysis of uniquely localizable sensor networks," in *Localization Algorithms and Strategies for Wireless Sensor Networks: Monitoring and Surveillance Techniques for Target Tracking*. IGI Global, 2009, pp. 146–173.

[34] I. Polik, T. Terlaky, and Y. Zinchenko, "SeDuMi: a package for conic optimization," in *IMA workshop on Optimization and Control, Univ. Minnesota, Minneapolis*. Citeseer, 2007.

[35] J. Lofberg, "YALMIP: A toolbox for modeling and optimization in matlab," in *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*. IEEE, 2004, pp. 284–289.

[36] P. Biswas, T.-C. Liang, K.-C. Toh, Y. Ye, and T.-C. Wang, "Semidefinite programming approaches for sensor network localization with noisy distance measurements," *IEEE transactions on automation science and engineering*, vol. 3, no. 4, pp. 360–371, 2006.