Algebraic Approach to Recovering Topological Information in Distributed Camera Networks

Edgar J. Lobaton, Parvez Ahammad, Shankar Sastry *

ABSTRACT

Camera networks are widely used for tasks such as surveillance, monitoring and tracking. In order to accomplish these tasks, knowledge of localization information such as camera locations and other geometric constraints about the environment (e.g. walls, rooms, and building layout) are typically considered to be essential. However, this information is not required for tasks such as estimating the topology of camera network coverage, or coordinate-free object tracking and navigation. In this paper, we propose a simplicial representation (called CN-Complex) that can be constructed from discrete local observations, and utilize this novel representation to recover topological information of the network coverage. We prove that our representation captures the correct topological information for coverage in 2.5D layouts, and demonstrate its utility in simulations as well as an experimental setup. Our proposed approach is particularly useful in the context of ad-hoc camera networks in indoor/outdoor urban environments with distributed but limited computational power and energy.

Categories and Subject Descriptors

I.1.2 [**Symbolic and Algebraic Manipulation**]: Algorithms

General Terms

Algorithm, Design, Theory

*E.J. Lobaton (`lobaton@eecs.berkeley.edu`) and S. Sastry are with the Electrical Engineering and Computer Sciences Dept., University of California, Berkeley, CA 94720, USA. P. Ahammad is with the Janelia Farm Research Campus, Howard Hughes Medical Institute, Ashburn, VA 20147, USA

Keywords

Camera Network Coverage, Distributed Sensing, Homology Theory, Topological Reconstruction

1. INTRODUCTION

Future generations of sensor networks are invariably going to include multiple types of sensors - including spatial sampling sensors such as cameras or active range scanners. Sensors like cameras will be the dominant consumers of bandwidth and power in such heterogeneous sensor networks. Thus, a clear understanding of the constraints (such as bandwidth consumption, power consumption, spatio-temporal sampling) posed by camera sensors in the context of computation and communication will play a critical role in defining the bounds for feasibility of performing certain tasks in a heterogeneous sensor network. In other words, such an understanding in the context of cameras could tell us whether our design of the heterogeneous network will be able to perform the designated task or not - and what conditions are necessary in order to perform such tasks.

Identification of the exact location of targets and objects in an environment is essential for many surveillance applications in the realm of sensor networks. However, there are situations in which the localization of the sensors is not known (e.g. unavailability of GPS, or ad-hoc network setup). A common approach to overcoming this challenge has been to determine the exact localization of the sensors and reconstruction of the surrounding environment. Nevertheless, we will provide evidence supporting the hypothesis that many of the tasks at hand may not require exact localization information.

Let us describe two potential scenarios in which some basic geometric information can aid in tracking and navigation for a large non-localized camera network:

1. Consider tracking of a target through an urban environment. In this scenario, it may be of interest to classify the path followed by the target. For example, it would be desirable for the network to specify whether the target went around a specific

landmark instead of returning a list of cameras in which the target was visible. In this work, we propose to identify paths that are homotopic to each other (i.e. that can be deformed continuously from one to another). This allows distinguishing between paths that go around a building clockwise or counter-clockwise without worrying about specific cameras visited. Note that this cannot be done by knowledge of pairwise connectivity between cameras (as in the case of so called connectivity/vision graphs).

2. A second scenario where topological information is useful is navigation through an urban environment. This task can be accomplished by making use of local target tracking and a set of directions such as where to turn right, and when to keep going straight. In this case, a general description of our surroundings and the target location is sufficient to guide us around obstacles.

One of the fundamental questions in the context of camera networks is whether a network is limited to perform only tasks that a single camera can perform but at a larger scale, or if the total network is "greater" than the sum of the parts. Imagine a camera network where no inter-relationship between the cameras is known. It is natural to ask what the spatial relationship between cameras is. For surveillance application in which multiple views are certainly useful, we investigate how object tracking information from multiple cameras can be aggregated and analyzed. A related and important question here is as to how we manage the processing and flow of data between the cameras. We note that all of these questions can be approached using knowledge of the topology of the coverage of the network. In particular, topology awareness makes it possible to design more efficient routing and broadcasting schemes as it is discussed by M. Li et al [10]. This knowledge in turn can also aid the control mechanism for more energy-efficient usage.
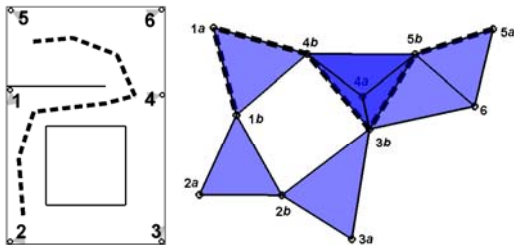

Figure 1: Physical layout (left) and simplicial representation (right) of an environment. In both cases we observe the paths of a target.

Figure 1 serves as a didactic tool to understand the in-

formation required for our approach to coordinate free tracking and navigation problems. Observe that the complete floor plan (left) and corresponding simplicial representation (right) serve an equivalent purpose. The simplicial representation allows us to track a target and navigate through the environment. Our goal in this context is to use the continuous observations from camera nodes to extract the necessary symbols to create this representation.

In this paper, we consider a camera network where each camera node can perform local computations, and extract symbolic/discrete observations to be transmitted for further processing. This conversion to symbolic representation alleviates the communication overhead for a wireless network. This is a significant benefit as self-localization algorithms can be computationally expensive and require the exchange of large volumes of data. We then use these discrete observations to build a model of the environment without any prior localization information of objects or the cameras themselves. Once such non-metric reconstruction of the camera network is accomplished, this representation can be used for tasks such as coordinate-free navigation, target-tracking, and path identification.

The rest of the discussion is as follows: Section 2 gives a brief discussion about different approaches to capturing topological information in sensor networks and the related work in this domain; sections 3 and 4 contain our main theoretical contributions defining the problems and the assumptions made for topological recovery of the camera network coverage; simulations and actual experiment are discussed in sections 5 and 6. The contributions of this paper include the introduction of formalized topology recovery problems by making explicit assumption about the environment model, and the utilization of topological information for tracking and navigation.

2. RELATED WORK

Finding the topology of a domain embedded in $\mathbb{R}^2$ is closely related to detecting holes. There has been much work on the detection and recovery of holes by topological methods for sensor networks, most of which considers symmetric coverage (explicitly or implicitly) or high enough density of sensors in the field. In particular, Vin de Silva and Ghrist [5] obtain the Rips complex based on the communication graph of the network and compute homologies using this representation. These methods assume some symmetry in the coverage of each sensor node (such as circular coverage), however, such assumptions are not valid for camera networks. Spatial sampling of plenoptic function [2] from a network of cameras is rarely independent and identically distributed. The notion of spatial coherence encountered in the context

of camera networks is not handled in traditional sensor network literature.

Connectivity between overlapping camera views by determining the correspondence models between cameras and extracting homography models has been approached by Stauffer and Tieu [18]. Cheng et al [4] build a vision graph in a distributed manner by exchanging feature descriptors from each camera view. In their work, each camera encodes a spatially well-distributed set of distinctive, approximately viewpoint-invariant feature points into a fixed-length "feature digest" that is broadcast throughout the network to establish correspondence between cameras. Yeo et al [19] utilize a random projection based framework to exchange compact feature descriptors in a rate-efficient manner to establish correspondence between various camera views.

Marinakis et al [12] work on finding connectivity between non-overlapping coverage of cameras by using only reports of detection and no description of the target. They use a Markov model for modeling the transition probabilities and minimize a functional using Markov Chain Monte Carlo Sampling. They also present a different formulation of the same problem with "timestamp free" observation with only ordering available (still no target description) [13]. Other approaches to solving the same problem with target identification have been explored by Zou et al [20]. Camera network with overlaps have been studied using the statistical consistency of the observation data by Makris et al[11]. Rahimi et al [17] describe a simultaneous calibration and tracking algorithm (with a networks of non-overlapping sensors) by using velocity extrapolation for a single target. Funiak et al [7] introduce a distributed algorithm for simultaneous localization and tracking with a set of overlapping cameras.

3. THE ENVIRONMENT MODEL

In this section the assumptions made for our problem are made explicit. Even though they may seem very restrictive, they are introduced in order to simplify the problem and facilitate the analysis.

3.1 The Problem in 2.5D

Our problem will be defined in terms of the detection of a target moving through an environment. For the sake of mathematical clarity, we first focus on the case of a single target moving through the environment. Let us start by describing our setup:

The Environment in 2.5D : We consider a domain in 3D with the following constraints:

- All objects and cameras in the environment will be within the space defined by the planes $z = 0$ (the "floor") and $z = h_{max}$ (the "ceiling").

- Objects in the environment consists of static "walls" erected perpendicular to our plane from $z = 0$ to $z = h_{max}$. The perpendicular projection of the objects to the plane $z = 0$ must have a piecewise linear boundary. Objects must enclose a non-zero volume.

Cameras in 2.5D : A camera α has the following properties:

- It is located at position o_α^{3D} with an arbitrary 3D orientation and a local coordinate frame Ψ_α^{3D}.

- Its **camera projection in 3D** , $\Pi_\alpha^{3D} : \mathcal{F}_\alpha \to \mathbb{R}^2$, is given by

$$\Pi_\alpha^{3D}(p) = (p_x/p_z, p_y/p_z),$$

where p is given in coordinate frame Ψ_α^{3D}, and $\mathcal{F}_\alpha \subset (\{(x, y, z) \,|\, z > 0\})$, referred to as the **field of view (FOV)** of the camera, is an open convex set such that its closure is a convex cone based at o_α^{3D}. The image of this mapping, i.e. $\Pi_\alpha^{3D}(\mathcal{F}_\alpha)$, will be called the **image domain** Ω_α^{3D}.

The Target in 2.5D : A target will have the following properties:

- The target will be a line segment perpendicular to the bounding planes of our domain which connects the points $(x, y, 0)$ to (x, y, h_t), where x and y are arbitrary and $h_t \leq h_{max}$ is the height of the target. The target is free to move along the domain as long as it does not intersect any of the objects in the environment.

- A target is said to be **detected** by camera α if there exists a point $p := (x, y, z)$ in the target such that $p \in \mathcal{F}_\alpha$ and $\overline{o_\alpha^{3D} p}$ does not intersect any of the objects in the environment.

Note that these assumptions may seem very restrictive, but they are satisfied by most camera networks in indoor and outdoor environments. Also, some of these choices in our model (such as the vertical line target and polygonal objects) are made in order to simplify our analysis. We will see that our methods work in real-life scenarios through our experiments.

The example in figure 2 shows a target and a camera with its corresponding FOV.

PROBLEM 1. **(2.5D Case):** *Given the camera and environment models in 2.5D , our goal is to obtain a representation that captures the topological structure of the* **detectable set for a camera network** *(i.e., the union of the sets in which a target is detectable by a camera). The construction of this representation should not rely on camera or object localization.*
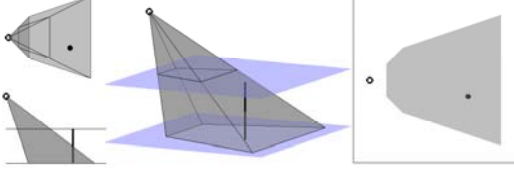
Figure 2: Mapping from 2.5D to 2D : A camera and its FOV are shown from multiple perspectives (left and middle), and its corresponding mapping to 2D (right). In 2.5D , the planes displayed bound the space that can be occupied by the target.

The formulation of the problem is very generic. We are choosing a simplicial representation because we are after a combinatorial representation that does not contain metric information. We are also after a distributed solution, i.e. processing information at local nodes.

3.2 Mapping from 2.5D to 2D

The structure of the detectable set for a camera network becomes clear through an identification of our 2.5D problem to a 2D problem. Since the target is constrained to move along the floor plane, it is possible to map our problem to a 2D problem. In particular:

- Cameras located at locations (x, y, z) are mapped to location (x, y) in the plane.

- Objects in our 2.5D domain are mapped to objects with piecewise linear boundaries in the plane.

- We can also do a simple identification between the FOV of a camera to a domain $\mathcal{D}_\alpha$ of a camera in 2D . A point (x, y) in the plane is in $\mathcal{D}_\alpha$ if the target located at that point intersects the FOV $\mathcal{F}_\alpha$. The set $\mathcal{D}_\alpha$ is the orthogonal projection (onto the xy-plane) of the intersection between $\mathcal{F}_\alpha$, and the space between $z \geq 0$ and $z \leq h_{target}$. Since the latter is an intersection of convex sets, and orthogonal projections preserve convexity, then $\mathcal{D}_\alpha$ is convex. We can also check that $\mathcal{D}_\alpha$ will be open.

- Also, we can give a 2D description of the coverage of a camera. A point (x, y) is in the coverage $\mathcal{C}_\alpha$ of camera α if the target located at (x, y) is detectable by the camera.

3.3 The Problem in 2D

We now proceed by characterizing our problem after mapping the original configuration from a 2.5D space to 2D . The following definitions are presented to formalize our discussion.

The Environment: The space under consideration is similar to the one depicted in figure 1 (left), where cameras are located in the plane, and only sets with **piecewise-linear** boundaries are allowed (including object and paths). We assume a finite number of objects in our environment.

Cameras: A camera object α is specified by: its **position** o_α in the plane; and an open convex domain $\mathcal{D}_\alpha$, referred to as the **camera domain**.

The camera domain $\mathcal{D}_\alpha$ can be interpreted as the set of points visible from camera α when no objects occluding the field of view are present. The convexity of this set will be essential for some of the proofs. Some examples of camera domains are shown in figure 4.

DEFINITION 1. *The subset of the plane occupied by the i-th **object**, which is denoted by $\mathcal{O}_i$, is a connected closed subset of the plane with non-empty interior and piecewise linear boundary. The collection $\{\mathcal{O}_i\}_{i=1}^{N_o}$, where $N_o < \infty$ is the number of objects in the environment, will be referred to as the **objects** in the environment.*

DEFINITION 2. *Given a camera α, a point $p \in \mathbb{R}^2$ is said to be **visible from camera** α if $p \in \mathcal{D}_\alpha$ and $\overline{o_\alpha p} \cap \left(\bigcup_{i=1}^{N_o} \mathcal{O}_i \right) = \emptyset$, where $\overline{o_\alpha p}$ is the line between the camera location o_α and p. The set of visible points is called the **coverage** $\mathcal{C}_\alpha$ of camera α.*

We consider the following problem:

PROBLEM 2. **(2D Case):** *Given the camera and environment models in 2D , our goal is to obtain a simplicial representation that captures the topological structure of the **coverage of the camera network** (i.e., the union of the coverage of the cameras). The construction of this representation should not rely on camera or object localization.*

OBSERVATION 1. *Note that the camera network coverage has the same homology (i.e. topological information) as the domain $(\mathbb{R}^2 - \bigcup O_i)$ if these two sets are homotopic (i.e., we can continuously deform one into the other).*

4. THE CN-COMPLEX

Our goal is the construction of a simplicial complex that will capture the homology of the union of camera coverages $\bigcup \mathcal{C}_\alpha$. One possible approach for accomplishing this task is to obtain the nerve complex (see appendix A) using the set of camera coverage $\{\mathcal{C}_\alpha\}$. However, this approach will only work for simple configurations without objects in the domain. An example illustrating our claim is shown in figure 3.

The reason figure 3 (right) does not capture the topological structure of the union of camera coverage is because the hypothesis of the Čech Theorem (see appendix
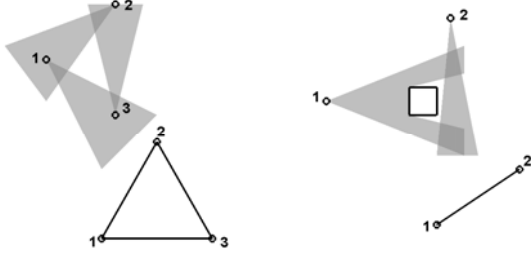
Figure 3: Nerve complexes obtained from the collection $\{\mathcal{C}_\alpha\}$. One complex captures the correct topological information (left) but the other does not (right).

A) is not satisfied (in particular, $\mathcal{C}_1 \cap \mathcal{C}_2$ is not contractible). From the physical layout of the cameras and the objects in the environment, it is clear how we can divide $\mathcal{C}_1$ in order to obtain contractible intersections. We are after a decomposition of the coverage that can be achieved without knowing the exact location of objects in the environment.

4.1   The Decomposition Theorem

Before we precede let us consider the following useful definitions:

DEFINITION 3. *Given the objects $\{\mathcal{O}_i\}_{i=1}^{N_o}$, a piecewise linear path $\Gamma : [0,1] \to \mathbb{R}^2$ is said to be* **feasible** *if $\Gamma([0,1]) \cap (\bigcup \mathcal{O}_i) = \emptyset$.*

DEFINITION 4. *Given camera α with camera domain $\mathcal{D}_\alpha$ and corresponding boundary $\partial \mathcal{D}_\alpha$, a line L_α is a* **bisecting line** *for the camera if:*

- *L_α goes through the camera location o_α.*

- *There exists a feasible path $\Gamma : [0,1] \to \mathbb{R}^2$ such that for any $\epsilon > 0$ there exists a δ such that $0 < \delta < \epsilon$, $\Gamma(0.5 - \delta) \in \mathcal{C}_n$, $\Gamma(0.5 + \delta) \notin \mathcal{C}_\alpha$, $\Gamma(0.5) \in L_\alpha$, and $\Gamma(0.5) \notin \partial \mathcal{D}_\alpha$.*

If we imagine a target traveling through the path Γ, we note that the last condition in the definition of a bisecting line identifies when an **occlusion event** is detected (i.e., the target transitions from visible to not visible, or vice versa). However, we will ignore the occlusion events due to the target leaving through the boundary of the camera domain $\mathcal{D}_\alpha$.

DEFINITION 5. *Let $\{L_{\alpha,i}\}_{i=1}^{N_L}$ be a finite collection of bisecting lines for camera α. Consider the set of adjacent cones in the plane $\{K_{\alpha,j}\}_{j=1}^{N_C}$ bounded by these lines, where $N_C = 2 \cdot N_L$, then the* **decomposition of** *$\mathcal{C}_\alpha$* **by lines** *$\{L_{\alpha,i}\}$ is the collection of sets*

$$\mathcal{C}_{\alpha,j} := K_{\alpha,j} \cap \mathcal{C}_\alpha.$$

Note that the decomposition of $\mathcal{C}_\alpha$ is not a **partition** since the sets $\mathcal{C}_{\alpha,j}$ are not necessarily disjoint.
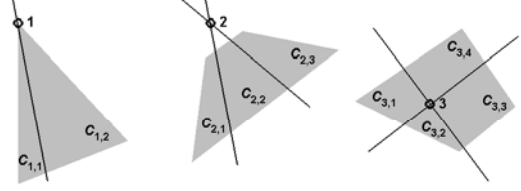

Figure 4: Three examples of camera domains $\mathcal{D}_\alpha$. Cameras can be inside or outside their domains. Our camera model spans projection models from perspective cameras to omni-directional cameras. Decompositions are shown for each set.

The construction of the **camera network complex** (CN-complex) is based on the identification of bisecting lines for the coverage of each individual camera. This construct will capture the correct topological structure of the union of coverage of the network.

Figure 5 displays examples of CN-complexes obtained after decomposing the coverage of each camera using their corresponding bisecting lines. The CN-complex captures the correct topological information, given that we satisfy the assumptions made for the model described in section 3. The following theorem (see appendix B for proof), states this fact.
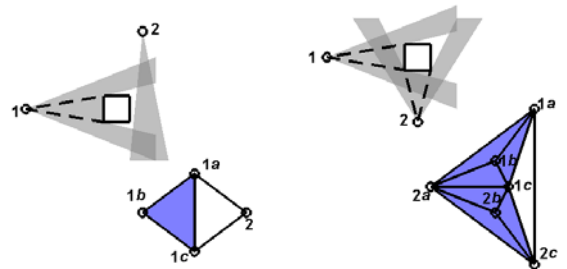

Figure 5: Examples of CN-complexes. In both cases, camera 1 is decomposed into three regions, each of which becomes a vertex in the complex.

THEOREM 1. **(Decomposition Theorem)**
Let $\{\mathcal{C}_\alpha\}_{\alpha=1}^N$ be a collection of camera coverage where each $\mathcal{C}_\alpha$ is connected and N is the number of cameras in the domain. Let $\{\mathcal{C}_{\alpha,k}\}_{(\alpha,k)\in A_D}$ be the collection of decomposed sets by all possible bisecting lines, where A_D is the set of indices in the decomposition. Then, any finite intersection $\bigcap_{(\alpha',k')\in A} \mathcal{C}_{\alpha',k'}$, where A is a finite set of indices, is contractible.

Hence, the hypothesis of the Čech Theorem is satisfied if we have connected coverage which are decomposed by all of their bisecting lines. This implies that computing

the homology of the CN-complex returns the appropriate topological information about the network coverage as a whole.

OBSERVATION 2. *Note that there are many ways to decompose a set in order to obtain subsets with contractible intersections. However, by using the bisecting lines, we ensure that the decomposition can be done locally (at each camera node) without knowledge of the physical structure of the environment.*

We note that the steps required to build the CN-complex are two-fold:

- Identify all bisecting lines and decompose each camera coverage.

- Determine which of the resulting sets intersect.

The first step makes sure that any intersection will be contractible. The second step allows us to find the simplices for our representation. These two steps can be completed in different ways which depend on the scenario under consideration. In sections 5 and 6, we illustrate the construction of the CN-complex for a very specific scenario.

4.2   From 2D to 2.5D

We can build the CN-complex by decomposing each camera coverage using its bisecting lines and determining which of the resulting sets intersect. However, a physical camera only has access to observations available in its image domain Ω^{3D}. Therefore, it is essential to determine how to find bisecting lines using information in the image domain.

We note that occlusion events occur when the target leaves the coverage $\mathcal{C}_\alpha$ of camera α along the boundary of the camera domain $\mathcal{D}_\alpha$ or along a bisecting line. We can verify that a target leaving through the boundary of $\mathcal{D}_\alpha$ will be detected in the image domain Ω_α^{3D} as having the target disappearing/appearing through the boundary of Ω_α^{3D}. If the target leaves $\mathcal{C}_\alpha$ through one of the bisecting lines, we will observe an occlusion event in the interior of Ω_α^{3D}. Note that bisecting lines in the 2D domain correspond to vertical planes in the 2.5D configuration, whose intersection with the FOV of the camera map to lines in Ω_α^{3D}. Hence, all that is required is to find the line segment in which an occlusion event takes place in the image domain. From an engineering point of view, this can be done by performing some simple image processing to find the edge along which target disappears/appears in an image. The result will be a decomposition of the image domain Ω_α^{3D} which will correspond to a decomposition of the camera coverage $\mathcal{C}_\alpha$. We also emphasize that these computations can be done locally at a camera node without any need to transmit information.

The problem of finding intersections of the sets for the 2D problem corresponds to having concurrent detections at corresponding cameras for the case of a single target in the environment. Finding overlap between these regions can be solved for the multiple-target case by using approaches such as the ones outlined in [12, 13, 20, 19, 4] in which correspondence and time correlation are exploited.

5.   SIMULATIONS IN 2D

We consider a scenario similar to the one shown in figure 1 (left) in which a wireless camera network is deployed and no localization information is available. Camera nodes will be assumed to have certain computational capabilities and they can communicate wirelessly with each other.

The assumptions for this particular simulation are:

The Environment in Simulation: The objects in the environment will have piecewise linear boundaries as described earlier. The location of the objects will be unknown. The location and orientation of the cameras is also unknown.

Cameras in Simulation: A camera α has the following properties:

- The domain $\mathcal{D}_\alpha$ of a camera in 2D will be the interior of a convex cone with field of view $\theta_\alpha < 180^o$. We use this model for simplicity in our simulations.

- A local camera frame Ψ_α^{2D} is chosen such that the range of the field of view is $[-\theta_\alpha/2, \theta_\alpha/2]$ when measured from the y-axis.

- Its **camera projection** $\Pi_\alpha^{2D} : \mathcal{D}_\alpha \to \mathbb{R}$, is given by

$$\Pi_\alpha^{2D}(p) = p_x/p_y,$$

  where p is given in coordinate frame Ψ_α^{2D}. The image of this mapping, i.e. $\Pi_\alpha^{2D}(\mathcal{D}_\alpha)$, will be called the **image domain** Ω_α^{2D}.

The Target in Simulation: A single point target is considered in order to focus on the construction of the complex without worrying about correspondence/identification of our target.

Throughout our simulations we will have the target moving continuously through the environment. At each time step the cameras compute their detections of the target and use their observations to detect bisecting lines. Observations at the regions obtained after decomposition using the bisecting lines are stored. These observations are then combined to determine intersections between the regions which become simplices in the CN-complex.

As mentioned before, the topology of the environment can be characterized in terms of its homology. In particular we will use betti numbers β_0 and β_1 (see appendix A). The β_0 number tells us the number of connected components in the coverage while β_1 gives the number of holes. The PLEX software package [1] is used for homology computations and corresponding betti numbers.
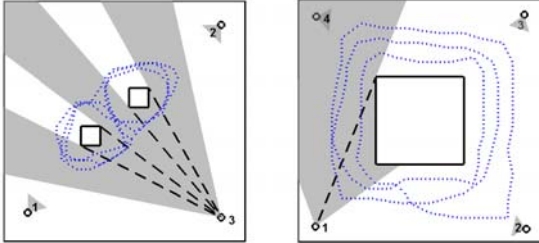

Figure 6: A layout with two objects (left) where $\mathcal{C}_3$ is shown. A circular hallway configuration (right) is shown. Dashed lines represent corresponding bisecting lines. Dotted curves represent the paths followed by the target.

Figure 6 (left) is a three-camera layout with two objects in their field of view. In this case, we observe three bisecting lines for camera 1, two for camera 2, and four for camera 3. Note that cameras 1 and 2 have different number of bisecting lines since there are not placed symmetrically in the diagram. The coverage $\mathcal{C}_3$ is decomposed into 5 regions, namely $\{\mathcal{C}_{3,a}, \mathcal{C}_{3,b}, \mathcal{C}_{3,c}, \mathcal{C}_{3,d} \text{ and } \mathcal{C}_{3,e}\}$. The list of maximal simplices obtained by our algorithm is: $[1a\ 1b\ 1c\ 1d]$, $[2a\ 2b\ 2c]$, $[3a\ 3b\ 3c\ 3d\ 3e]$, $[1a\ 1b\ 2c\ 3c]$, $[1d\ 2a\ 3c]$, $[2a\ 2b\ 3a]$, $[1a\ 2b\ 2c\ 3a]$, $[1a\ 2c\ 3a\ 3b]$, $[1a\ 2c\ 3b\ 3c]$, $[1a\ 2c\ 3c\ 3d]$, $[1a\ 1b\ 1c\ 2c\ 3d\ 3e]$, $[1c\ 1d\ 3e]$ and $[1d\ 2a\ 3e]$. The homology computations returned betti numbers: $\beta_0 = 1$ and $\beta_1 = 2$. This agrees with having a single connected component for the network coverage and two objects inside the coverage of the cameras.

In figure 6 (right) we observe similar results for a configuration that can be interpreted as a hallway in a building floor. There is a single bisecting line for all cameras. Our algebraic analysis returns $\beta_0 = 1$ and $\beta_1 = 1$. The latter identifies a single hole corresponding to the loop formed by the hallway structure. The list of maximal simplices recovered by our algorithm: $[3b\ 4a\ 4b]$, $[2b\ 3a\ 3b]$, $[1b\ 2a\ 2b]$, $[1a\ 1b\ 4b]$.

6. EXPERIMENTATION

In order to demonstrate how the mathematical tools described in the previous sections can be applied to a real wireless sensor network, we setup an experiment tracking a robot in a simple maze. Figure 1 shows the layout to be used. We placed a sensor network consisting of CITRIC camera motes [6] at several locations in our maze and let a robot navigate through the environment. The CN-complex is constructed for this particular coverage and used for tracking in this representation. Homology computations are performed using the PLEX software package [1].

Time synchronization is required in order to determine overlaps between the different camera regions. The Flooding Time Synchronization Protocol (FTSP) [14] was used for this purpose.

At each camera node, background subtraction is performed at each frame. Once a target is detected, we perform further processing to detect bisecting lines (as shown in figure 7). However, note that the bisecting line processing occurs sparsely and hence power consumption is mostly due to background subtraction. Statistics on the power consumption for the CITRIC platform can be found in [6]. Note that the information extracted from each camera node is just a decomposition of the image domain with a list of times at which detections were made.

For our experiments the camera motes were capable of processing grayscale images at 4 frames per second at a resolution of 320×240 pixels. Symbolic information was then extracted and transmitted at a rate of 1 packet of 100 bytes every 10 seconds. We transmitted regularly even when there were no observations to transmit. If raw image data (without any compression) was to be streamed over the network, this would correspond to about 300 $kBytes/s$ of data from a single mote. Instead, transmitting symbolic information in our experiment only accounts for 10 $Bytes/s$.
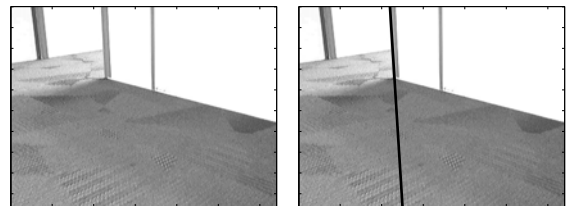

Figure 7: View of camera 5 from the layout in figure 1 before (left) and after (right) a bisecting line is found.

The complex is built by combining all local information from the camera motes. Each camera mote transmits the history of its detections wirelessly to a central computer that creates the CN-complex. The resulting complex contains the maximal simplices: $[1a\ 1b\ 4b]$, $[1b\ 2a\ 2b]$, $[2b\ 3a\ 3b]$, $[3b\ 4a\ 4b\ 5b]$, $[3b\ 5b\ 6]$, and $[5a\ 5b\ 6]$. A pictorial representation of the complex is shown in figure 8 (right plots).

As mentioned earlier, this representation can then be used for tracking and navigation without actual metric reconstruction of the environment. Figure 8 shows a set of recorded paths for our robot. By determining which

simplices are visited by the robot's path we can extract a path in the complex as shown by the dashed path in the complexes of figure 8. The main advantage of this representation is that the path in the complex gives a global view of the trajectory of the robot, while local information can be extracted from single camera views.
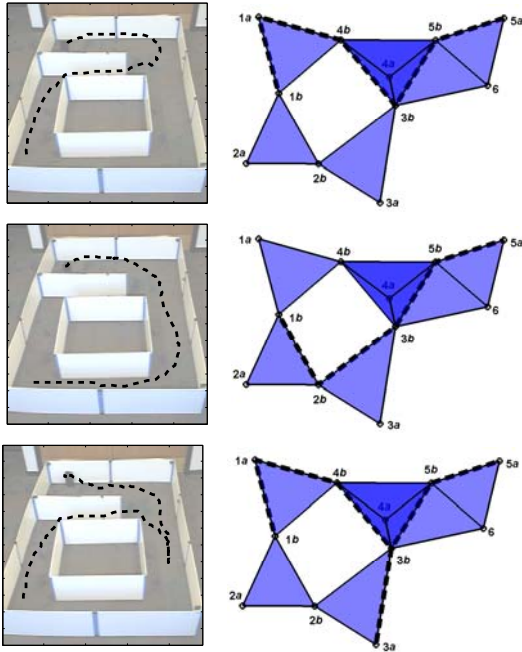

Figure 8: Paths in the maze (shown in dashed lines): In the physical layout (left), and in the CN-complex (right). These paths can be compared by using the algebraic topological tools covered in appendix A.

It is possible to identify paths in the simplicial representation that are homotopic (i.e., that can be continuously deformed into one another). The tools required for these computations are already available to us from appendix A. In particular, by taking two paths that start and end at the same locations forming a loop, we can verify that they are homotopic if they form the boundary of some combination of simplices. Equivalently, since a closed loop σ is just a collection of edges in C_1, we need to check whether the loop σ is in B_1 (i.e., in the range of ∂_2). This is just a simple algebraic computation. By putting the top and middle paths from figure 8 together we note that the resulting loop is not in the range of ∂_2 (i.e., they are not homotopic). On the other hand, the top and bottom paths can be easily checked to be homotopic.

7. SUMMARY AND DISCUSSION

In this paper, an algebraic representation of a camera network coverage is obtained through the use of discrete observations from each camera node. The mathematical tools used for this purpose are those of algebraic topology. In particular, we showed that given enough observations our model does capture the correct topological information.

The experiment using wireless camera motes illustrates how our representation can be used to track and compare paths in a wireless camera network without any metric information. For coordinate-free navigation, our representation can give an overall view of how to arrive at a specific location, and the transitions between simplices can be accomplished in the physical space by local visual feedback from single camera views. Using this proposed model allows for local processing at each node and minimal wireless communication. A list of times at which occlusion events were observed is all that needs to be transmitted. Also, all algebraic computations can be performed using integer operations as described in [9], which opens the doors to implementation on platforms with low-computational power. The homology computations in the experiment are done in a centralized fashion, however, distributed algorithms such as the ones introduced by A. Muhammad and A. Jadbabaie [15] can be used.

The approach described above can be extended to more complex environments with stairs, windows and building with multiple levels. In the future, we hope to extend this work to handle multiple targets and detection errors. In our experiments, simplices are drawn every time concurrent detections are observed. However, false detections can cause the discovery of incorrect simplices. This can be solved by constructing a non-deterministic complex that assigns probabilities to simplices. This will be the focus of our future research.

Acknowledgements

8. REFERENCES

[1] PLEX: A sytem for computational homology, Oct 2008.
http://comptop.stanford.edu/programs/plex.html.
[2] E. H. Adelson and J. Bergen. The plenoptic function and the elements of early vision. *M. Landy and J. A. Movshon, (eds) Computational Models of Visual Processing*, pages 3–20, 1991.
[3] R. Bott and L. Tu. *Differential Forms in Algebraic Topology.* Springer, 1995.
[4] Z. Cheng, D. Devarajan, and R. Radke. Determining vision graphs for distributed camera networks using feature digests. *EURASIP Journal on Advances in Signal Processing*, 2007, 2007.
[5] V. de Silva and R. Ghrist. Coordinate-free coverage in sensor networks with controlled boundaries via homology. *The International Journal of Robotics Research*, 25:1205 – 1221, 2006.
[6] P. C. et al. CITRIC: A low-bandwidth wireless camera network platform. In *Third ACM/IEEE International Conference on Distributed Smart Cameras*, 2008.

[7] S. Funiak, C. Guestrin, M. Paskin, and R. Sukthankar. Distributed localization of networked cameras. In *Proceedings of the fifth international conference on Information processing in sensor networks*, 2006.

[8] A. Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.

[9] T. Kaczynski, K. Mischaikow, and M. Mrozek. *Computational Homology*. Springer, 2003.

[10] M. Li and B. Yang. A survey on topology issues in wireless sensor network. In *Proceedings of the International Conference on Wireless Networks*, 2006.

[11] D. Makris, T. Ellis, and J. Black. Bridging the gaps between cameras. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004.

[12] D. Marinakis and G. Dudek. Topology inference for a vision-based sensor network. In *Proceedings of the Second Canadian Conference on Computer and Robot Vision*, 2005.

[13] D. Marinakis, P. Giguere, and G. Dudek. Learning network topology from simple sensor data. In *Proceedings of the twentieth Canadian Conference on Artificial Intelligence*, 2007.

[14] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi. The flooding time synchronization protocol. In *Proceedings of the Second international conference on Embedded networked sensor systems*, 2004.

[15] A. Muhammad and A. Jadbabaie. Decentralized computation of homology groups in networks by gossip. In *Proceedings of the American Control Conference*, 2007.

[16] J. Munkres. *Topology*. Prentice Hall, 2nd edition, 2000.

[17] A. Rahimi, B. Dunagan, and T. Darrell. Simultaneous calibration and tracking with a network of non-overlapping sensors. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–187–I–194, 2004.

[18] C. Stauffer and K. Tieu. Automated multi-camera planar tracking correspondence modeling. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003.

[19] C. Yeo, P. Ahammad, and K. Ramchandran. Rate-efficient visual correspondences using random projections. In *Proceedings of IEEE International Conference on Image Processing*, October 2008.

[20] X. Zou, B. Bhanu, B. Song, and A. Roy-Chowdhury. Determining topology in a distributed camera network. In *IEEE International Conference on Image Processing*, 2007.

APPENDIX

A. MATHEMATICAL BACKGROUND

In this section we cover the concepts from algebraic topology that will be used throughout this paper. This section contains material adapted from [8, 5] and it is not intended as a formal introduction to the topic. For a proper introduction to the topic, the reader is encouraged to read [16, 9, 8].

A.1 Simplicial Homology

DEFINITION 6. *Given a collection of vertices V we define a k-simplex as a set $[v_1 \ v_2 \ v_3 \ \ldots \ v_{k+1}]$ where $v_i \in V$ and $v_i \neq v_j$ for all $i \neq j$. Also, if A and B are simplices and the vertices of B form a subset of the vertices of A, then we say that B is a face of A.*

DEFINITION 7. *A finite collection of simplices in $\mathbb{R}^n$ is called a simplicial complex if whenever a simplex lies in the collection then so does each of its faces.*

DEFINITION 8. *The nerve complex of a collection of sets $\mathcal{S} = \{S_i\}_{i=1}^N$, for some $N > 0$, is the simplicial complex where vertex v_i corresponds to the set S_i and its k-simplices correspond to non-empty intersections of $k + 1$ distinct elements of $\mathcal{S}$.*

The following statements define some algebraic structures using these simplices.

DEFINITION 9. *Let $\{s_i\}_{i=1}^N$ (for some $N > 0$) be the k-simplices of a given complex. Then, the group of k-chains C_k is the free abelian group generated by $\{s_i\}$. That is,*

$$\sigma \in C_k \quad \text{iff} \quad \sigma = \alpha_1 s_1 + \alpha_2 s_2 + \cdots \alpha_N s_N$$

for some $\alpha_i \in \mathbb{Z}$. If there are no k-simplices, then $C_k := 0$. Similarly, $C_{-1} := 0$.

DEFINITION 10. *Let the boundary operator ∂_k applied to a k-simplex s, where $s = [v_1 \ v_2 \ \cdots \ v_{k+1}]$, be defined by:*

$$\partial_k s = \sum_{i=1}^{k+1} (-1)^{i+1} [v_1 \ v_2 \ \cdots \ v_{i-1} \ v_{i+1} \ \cdots \ v_k \ v_{k+1}],$$

and extended to any $\sigma \in C_k$ by linearity.
A k-chain $\sigma \in C_k$ is called a cycle if $\partial_k \sigma = 0$. The set of k-cycles, denoted by $\mathbf{Z_k}$, is the $\ker \partial_k$ and forms a subgroup of C_k. That is,

$$Z_k := \ker \partial_k.$$

A k-chain $\sigma \in C_k$ is called a boundary if there exists $\rho \in C_{k+1}$ such that $\partial_{k+1} \rho = \sigma$. The set of k-boundaries, denoted by $\mathbf{B_k}$, is the image of ∂_{k+1} and it is also a subgroup of C_k. That is,

$$B_k := \text{im } \partial_{k+1}.$$

Even further, we can check that $\partial_k(\partial_{k+1}\sigma) = 0$ for any $\sigma \in C_{k+1}$, which implies that B_k is a subgroup of Z_k.

Observe that the boundary operator ∂_k maps a k-simplex to its $(k-1)$-simplicial faces. Further, the set of edges that form a closed loop are exactly what we denote by the group of 1-cycles. We will be interested in finding out holes in our domains; that is, cycles that cannot be obtained from boundaries of simplices in a given complex. This observation motivates the definition of the homology groups.

DEFINITION 11. *The k-th homology group is the quotient group*

$$H_k := Z_k / B_k.$$

The homology of a complex is the collection of all homology groups. The rank of H_k, denoted the k-th betti number β_k, gives us a coarse measure of the number of holes. In particular, β_0 is the number of connected components and β_1 is the number of loops that enclose different "holes" in the complex.

A.2 Example

In figure 9 we observe a collection of triangular shaped sets labeled from 1 to 5. The nerve complex is obtained by labeling the 0-simplices (i.e., the vertices) in the same way as the sets. The 1-simplices (i.e., the edges in the pictorial representation) correspond to pairwise intersection between the regions. The 2-simplex correspond to the intersection between triangles 2, 4 and 5.

For the group of 0-chains C_0, we can identify the simplices $\{[1], [2], [3], [4], [5]\}$ with the column vectors $\{v_1, v_2, v_3, v_4, v_5\}$, where $v_1 = [1, 0, 0, 0, 0]^T$ and so on.
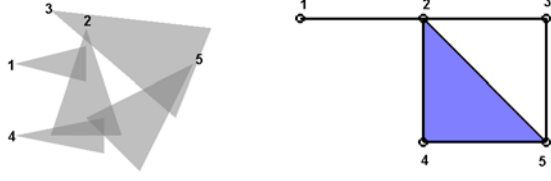
Figure 9: A collection of sets (left) and corresponding nerve complex (right). The complex is formed by simplices: [1], [2], [3], [4], [5], [1 2], [2 3], [2 4], [2 5], [3 5], [4 5] and [2 4 5].

For C_1, we identify $\{[1\ 2], [2\ 3], [2\ 4], [2\ 5], [3\ 5], [4\ 5]\}$ with the column vectors $\{e_1, e_2, e_3, e_4, e_5, e_6\}$, where we define $e_1 = [1, 0, 0, 0, 0, 0]^T$ and so on.

Similarly for C_2, we identify $[2\ 4\ 5]$ with $f_1 = 1$.

As we mentioned before, ∂_k is the operator that maps a simplex $\sigma \in C_k$ to its boundary faces. For example, we have:

$$\partial_2[2\ 4\ 5] = [4\ 5] - [2\ 5] + [2\ 4] \quad \text{iff} \quad \partial_2 f_1 = e_6 - e_4 + e_3,$$

$$\partial_1[2\ 4] = [4] - [2] \quad \text{iff} \quad \partial_1 e_3 = v_4 - v_2.$$

That is, ∂_k can be expressed in matrix form as:

$$\partial_1 = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \quad , \quad \partial_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \\ 0 \\ 1 \end{bmatrix}.$$

Since $C_{-1} = 0$,

$$H_0 = Z_0/B_0 = ker\ \partial_0/im\ \partial_1 = C_0/im\ \partial_1.$$

We can verify that

$$\beta_0 = dim(H_0) = 1.$$

Hence, we recover the fact that we have only one connected component in the diagram of figure 9. Similarly, we can verify that

$$\beta_1 = dim(H_1) = dim(Z_1/B_1) = dim(ker\ \partial_1/im\ \partial_2) = 1,$$

which tells us that the number of holes in our complex is 1. Also, $H_k = 0$ for $k > 1$ (since $C_k = 0$).

A.3 Čech Theorem

Now we introduce the Čech Theorem which has been used in the context of sensor networks with unit-disk coverage [5] and has been proved in [3]. Before we proceed further, we will require the following definition:

DEFINITION 12. *Given two spaces X and Y, a **homotopy** between two continuous functions $f_0 : X \to Y$ and $f_1 : X \to Y$ is a continuous 1-parameter family of continuous functions $f_t : X \to Y$ for $t \in [0,1]$ connecting f_0 to f_1.*

DEFINITION 13. *Two spaces X and Y are said to be of the same **homotopy type** if there exist functions $f : X \to Y$ and $g : Y \to X$ with $g \circ f$ homotopic to the identity map on X and $f \circ g$ homotopic to the identity map on Y.*

DEFINITION 14. *A set X is **contractible** if the identity map on X is homotopic to a constant map.*

In other words, two functions are homotopic if we can continuously deform one into the other. Also, a space is contractible if we can continuously deform it to a single point. It is known that homologies are an **invariant** of homotopy type; that is, two spaces with the same homotopy type will have the same homology groups.

THEOREM 2. (**Čech Theorem**) *If the sets $\{S_i\}_{i=1}^N$ (for some $N > 0$) and all nonempty finite intersections are contractible, then the union $\bigcup_{i=1}^N S_i$ has the homotopy type of the nerve complex.*

That is, given that the required conditions are satisfied, the topological structure of the union of the sets is captured by the nerve. We observe that in figure 9 all of the intersections are contractible. Therefore, we can conclude that the extracted nerve complex has the same homology as the space formed by the union of the triangular regions.

B. PROOF OF THEOREM 1

Throughout this section we consider a finite set of cameras indexed by $\alpha \in \{1, 2, 3 \cdots N_c\}$ with corresponding domains $\mathcal{D}_\alpha$ and coverages $\mathcal{C}_\alpha$. Each camera coverage is decomposed by all possible bisecting lines $\{L_{\alpha,i}\}$. The collection $\{\mathcal{C}_{\alpha,j}\}$ is the result of this decomposition, where $\mathcal{C}_{\alpha,j} := \mathcal{C}_\alpha \cap K_{\alpha,j}$ and $K_{\alpha,j}$ is the convex cone resulting from decomposing the plane using the lines $\{L_{\alpha,i}\}$ (see definition 4).

OBSERVATION 3. *It may be useful for the reader to think of the set $\mathcal{C}_n$ (the visible set after object occlusions have been removed) as the intersection of a convex set (i.e., the camera domain) with a star convex set (due to visibility from o_α).*

OBSERVATION 4. *The number of bisecting lines for a given camera in our environment is finite since we are considering finite number of objects in the coverage with piecewise linear boundaries.*

DEFINITION 15. *The **line segment joining points** p and q is denoted by $\overline{pq}$. The **line passing through points** p and q is denoted by $L(p, q)$.*

DEFINITION 16. *The **triangle formed by points** a, b and $c \in \mathbb{R}^2$ is the convex hull of these three points and it is denoted $\Delta_{a,b,c}$.*

LEMMA 1. *Given that $o_\alpha, p \in \mathcal{C}_\alpha$ then $\overline{o_\alpha\,p} \in \mathcal{C}_\alpha$.*

PROOF. Since, o_α and $p \in \mathcal{C}_\alpha \subset \mathcal{D}_\alpha$, then $\overline{o_\alpha\,p} \subset \mathcal{D}_\alpha$ due to convexity of $\mathcal{D}_\alpha$. Let $r \in \overline{o_\alpha\,p}$. If r is not visible then $\overline{o_\alpha\,r} \cap \bigcup \mathcal{O}_i \neq \emptyset$ (where $\{\mathcal{O}_i\}$ is the collection of objects in the environment). However, this implies that $\overline{o_\alpha\,p} \cap \bigcup \mathcal{O}_i \neq \emptyset$. Hence, we conclude that p is not visible, which is a contradiction. Therefore, r must be visible. Since r was arbitrary then $\overline{o_\alpha\,p}$ is visible. $\square$

LEMMA 2. *Given that $p, q \in \mathcal{C}_\alpha$ with*

$$L(p, o_\alpha) = L(q, o_\alpha),$$

then $\overline{pq} \in \mathcal{C}_\alpha$. That is, if p and q are visible and are in the same line of sight, then the line joining them is visible too.

PROOF. This follows from the definition of $\mathcal{C}_\alpha$ and the domain of a camera $\mathcal{D}_\alpha$. We know that $\mathcal{D}_\alpha$ is convex, so $\overline{pq} \subset \mathcal{D}_\alpha$ since $p, q \in \mathcal{C}_\alpha \subset \mathcal{D}_\alpha$.

From our assumption $L(p, o_\alpha) = L(q, o_\alpha)$, it is possible to conclude that for $r \in \overline{pq}$ then $r \in \mathcal{D}_\alpha$, and $r \in \overline{o_\alpha\,p}$ or $r \in \overline{o_\alpha\,q}$. Basically, there are only two cases, both p and q on the same side of o_α or on opposites sides. Either way, r must be in $\overline{o_\alpha\,p}$ or $\overline{o_\alpha\,q}$.

Without loss of generality, assume $r \in \overline{o_\alpha\,p}$. If r was not visible, the

$$\overline{o_\alpha\,r} \cap \bigcup \mathcal{O}_i \neq \emptyset$$

(where $\{\mathcal{O}_i\}$ is the collection of sets representing the objects in the space). This implies that

$$\overline{o_\alpha\,p} \cap \bigcup \mathcal{O}_i \neq \emptyset,$$

since $\overline{o_\alpha\,r} \subset \overline{o_\alpha\,p}$. This implies that $p \notin \mathcal{C}_\alpha$ which is a contradiction. Therefore, r must be visible too. $\square$
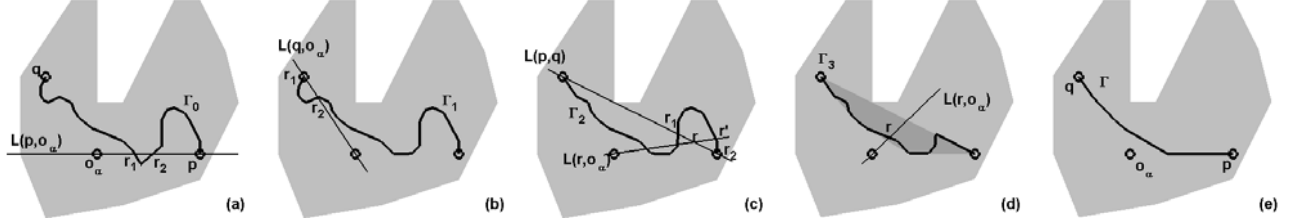
Figure 10: Steps of the construction of a monotone convex path for lemma 6.

LEMMA 3. *Given a closed path* $\Gamma([0,1]) \subset \mathcal{C}_\alpha$, *then the space enclosed by* Γ *is also in* $\mathcal{C}_\alpha$.

PROOF. Let $\mathcal{R}$ be the enclosed area by the path Γ. Since $\Gamma : [0,1] \to \mathbb{R}^2$ is bounded, then

$$\exists M > 0 \text{ such that } ||\Gamma(t) - o_\alpha|| < M,$$

where o_α is the location of camera α. Hence,

$$r \notin \mathcal{R} \text{ if } ||r - o_\alpha|| > M.$$

Also, if a point r' is connected to $r \notin \mathcal{R}$ through a path γ that does not cross Γ, then $r' \notin \mathcal{R}$.

Let $p \in \mathcal{R}$ and define

$$\mathcal{L} := L(p, o_\alpha) \cap \Gamma([0,1])$$

(i.e., points in Γ and in the line passing through p and o_α), then there must be points $q_1, q_2 \in \mathcal{L}$ such that and $p \in \overline{q_1 q_2}$. Otherwise, there would exist a point $r \in L$ with $||r - o_\alpha|| > M$ (i.e., $r \notin \mathcal{R}$) such that $\overline{rp}$ does not intersect $\Gamma([0,1])$. This implies $p \notin \mathcal{R}$ which is a contradiction. Therefore, $p \in \overline{q_1 q_2}$.

Next, we consider three cases:

- Assume $q_1 \neq o_\alpha$ and $q_2 \neq o_n$. Since $q_1, q_2 \in \Gamma([0,1]) \subset \mathcal{C}_n$ with $L(q_1, o_\alpha) = L(q_2, o_\alpha)$, then $p \in \overline{q_1 q_2} \subset \mathcal{C}_n$ by lemma 2 (which makes p visible).

- Assume $q_1 \neq o_\alpha$ and $q_2 = o_\alpha$. Then $p \in \overline{o_\alpha q_2} \subset \mathcal{C}_\alpha$ by lemma 1.

- Assume $q_1 = q_1 = o_\alpha$. Then, $p = o_\alpha \in \mathcal{C}_\alpha$.

In all cases p is visible, and since p was arbitrary we conclude that $\mathcal{R}$ is visible. $\square$

The previous lemmas are also true if we replace $\mathcal{C}_\alpha$ by the set $\mathcal{C}_{\alpha,j}$ resulting from a decomposition of the coverage. The reason why it works is because we can think of $\mathcal{C}_{\alpha,j}$ as being the coverage of a camera with a domain

$$\mathcal{D}_{\alpha,j} := \mathcal{D}_\alpha \cap K_{\alpha,j},$$

where $K_{\alpha,j}$ is the corresponding convex cone that generates the region $\mathcal{C}_{\alpha,j}$. This new domain is still convex which is the property used in the previous lemmas. However, note that this $\mathcal{D}_{\alpha,j}$ is not open.

LEMMA 4. *Every connected component of* $\bigcap_{(\alpha,j) \in A} \mathcal{C}_{\alpha,j}$, *where A is a finite set of indices, is simply connected.*

PROOF. Let Γ be a closed loop in $\bigcap_{(\alpha,j) \in A} \mathcal{C}_{\alpha,j}$. By the previous lemma, the space enclosed by Γ is inside $\mathcal{C}_{\alpha,j}$ for all $(\alpha,j) \in A$. $\square$

DEFINITION 17. *Let* $\Gamma : [0,1] \to \mathbb{R}^2$ *be a path connecting points p to q (i.e. $\Gamma(0) = p$ and $\Gamma(1) = q$). We define the* **region enclosed by** Γ, *denoted by $\mathcal{R}(\Gamma)$, to be the region enclosed by the set $\Gamma([0,1]) \cup \overline{pq}$.*

DEFINITION 18. *A path* $\Gamma : [0,1] \to \mathbb{R}^2$ *connecting points p and q is said to be a* **convex path** *if $\mathcal{R}(\Gamma)$ is convex.*

DEFINITION 19. *A non-intersecting path* $\Gamma : [0,1] \to \mathbb{R}^2$ *is* **monotone with respect to camera** α *if for any $p \in S_\alpha^1$, where S_α^1 is the unit circle centered at o_α, we have that $\Gamma([0,1]) \cap L(p, o_\alpha)$ has a single connected component.*

LEMMA 5. *Let $\mathcal{R}$ be a bounded convex set contained between the lines $L(p, o_\alpha)$ and $L(q, o_\alpha)$, where p and $q \in \mathcal{R}$. Then, either $\overline{pq}$ is the only path in $\mathcal{R}$ joining p to q, or there are exactly two distinct images of monotone paths connecting p to q (only intersecting at the end points), which form the boundary of $\mathcal{R}$.*
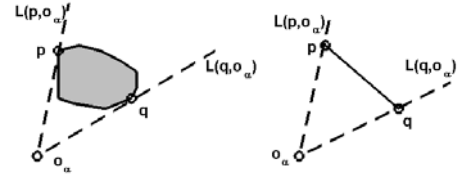
The figure above illustrates the results from the previous lemma.

LEMMA 6. *Given that $\mathcal{C}_\alpha$ is connected with p and $q \in \mathcal{C}_\alpha$, then there exists a path Γ connecting these points that is convex and monotone with respect to camera α with $\Gamma([0,1]) \subset \mathcal{C}_\alpha \cap \Delta_{p,q,o_\alpha}$.*

PROOF. We present an outline of the proof of this result. Let p and $q \in \mathcal{C}_\alpha$, where $\mathcal{C}_\alpha$ is connected.

The reader may be tempted to try the path $\overline{p o_\alpha} \cup \overline{o_\alpha q}$. However, we are not assuming $o_\alpha \in \mathcal{C}_\alpha$. Our proof takes care of this case too.

Since $\mathcal{C}_\alpha$ is connected then there exists a path Γ_0 that connects p to q with $\Gamma_0([0,1]) \subset \mathcal{C}_\alpha$. We illustrate this in the diagram in figure 10 (a) in which the gray region corresponds to the coverage under consideration.

Our first objective will be to construct a path that is contained within Δ_{p,q,o_α}.

We start with path Γ_0 and consider the line $L(p, o_\alpha)$ (see figure 10 (a)). This line will intersect the Γ_0 at points $\{r_k\}$. By lemma 2, we know that the line segments between them are visible, so we can construct path Γ_1 (as shown in figure 10 (b)) which does not cross $L(p, o_\alpha)$.

Next, we consider the intersections between $L(q, o_\alpha)$ and Γ_1 (see figure 10 (b)). As in the previous case, we can construct a path Γ_2 which does not cross $L(q, o_\alpha)$.

If we consider the line $L(p, q)$, then it will intersect the line Γ_2 at points $\{r_k\}$ (see figure 10 (c)). Consider a segment of Γ_2 that is outside of the triangle Δ_{p,q,o_α}, which intersects $L(p, q)$ at r_1 and r_2. For any $r \in \overline{r_1 r_2}$, we see that $r \in \mathcal{D}_\alpha$ since $r_k \in \mathcal{D}_\alpha$ and $\mathcal{D}_\alpha$ is convex. Also, there exists a point $r' \in L(r, o_\alpha) \cap \Gamma_2([0,1])$ which is further away from o_α than r. Otherwise, the line segment in Γ between r_1 and r_2 would not be outside the Δ_{p,q,o_α}. Therefore, if r was not visible then r' would not be visible which is a contradiction. Hence, r must be visible.

This implies that we can connect r_1 to r_2 by the line segment $\overline{r_1 r_2}$ and construct path Γ_3 which is inside Δ_{p,q,o_α}.

In order to make Γ_3 into a convex path, we take the convex hull of Γ_3 and by lemma 5 we know that there are at most two monotone paths to choose from (see figure 10 (d)). We choose the path Γ that is closest to o_α. Clearly Γ is convex. We can see that Γ is visible since for any line $L(r, o_\alpha)$ for $r \in \Gamma_3([0,1])$, the line will have to intersect Γ at some location s closer to o_α than r.

This process yields the desired monotone and convex path Γ (see figure 10 (e)) which images is in $\mathcal{C}_\alpha \cap \Delta_{p, o_\alpha, q}$. $\square$

LEMMA 7. *Given that $\mathcal{C}_\alpha$ is connected with p and $q \in \mathcal{C}_{\alpha, j}$ for some j, then there exists a path Γ connecting these points that is convex and monotone with respect to camera α with $\Gamma([0,1]) \subset \mathcal{C}_{\alpha,j} \cap \Delta_{p,q,o_\alpha}$.*

PROOF. Since p and $q \in \mathcal{C}_{\alpha,j}$, then p and $q \in C_\alpha \cap K_{\alpha,j}$. By the previous lemma, we know that there exists a path Γ such that $\Gamma([0,1]) \subset \mathcal{C}_\alpha$. Note that $\Gamma([0,1])$ is inside the cone formed by the lines $L(p, o_\alpha)$ and $L(q, o_\alpha)$ by construction. This cone must be contained within $K_{\alpha,j}$, otherwise p and q could not be in $K_{\alpha,j}$. Therefore, $\Gamma([0,1]) \subset K_{\alpha,j} \cap \mathcal{C}_\alpha = \mathcal{C}_{\alpha,j}$. $\square$

LEMMA 8. *Let $\Gamma : [0,1] \to \mathbb{R}^2$ be a feasible monotone path connecting p and $q \in \mathcal{C}_\alpha$ with $\Gamma([0,1]) \subset \mathcal{D}_\alpha$ for some camera α. If an object $\mathcal{O}$ is within the region enclosed by $\overline{o_\alpha p} \cup \Gamma([0,1]) \cup \overline{q o_\alpha}$ then there exists a bisecting line L passing through a point in Γ that does not intersect $L(p, o_\alpha)$ and $L(q, o_\alpha)$ (not including these lines).*

PROOF. For simplicity we just give an outline of this proof.

Since $\Gamma([0,1]) \subset \mathcal{D}_\alpha$, we know that no point in Γ will be in the boundary of $\mathcal{D}_\alpha$ since $\mathcal{D}_\alpha$ is open.

Since $\overline{o_\alpha p} \cup \Gamma([0,1]) \cup \overline{q o_\alpha}$ encloses an object, there exists a transition between having a visible and a not-visible point in the path (i.e. an occlusion event). This is guaranteed since at least a point in the path is visible, and not all the points can be visible due to the object $\mathcal{O}$.

Assume that the transition event occurs in $L(p, o_\alpha)$ or $L(q, o_\alpha)$ at some point $r \in \Gamma([0,1])$ and nowhere else. Without loss of generality assume that $\overline{rp} \subset \Gamma([0,1])$ (due to monotonicity of path). The object would have to occlude r too (since objects are closed). Then either the path is not feasible or p is not visible which contradicts our assumption. Therefore, a transition must occur at some other point along Γ and not in these lines. $\square$

THEOREM 3. **(Decomposition Theorem)** *Let $\{\mathcal{C}_\alpha\}_{\alpha=1}^N$ be a collection of camera coverages where each $\mathcal{C}_\alpha$ is connected and N is the number of cameras in the domain. Let $\{\mathcal{C}_{\alpha,k}\}_{(\alpha,k) \in A_D}$ be the collection of decomposed sets by all possible bisecting lines, where A_D is the set of indices in the decomposition. Then, any finite intersection $\bigcap_{(\alpha',k') \in A} \mathcal{C}_{\alpha',k'}$, where A is a finite set of indices, is contractible.*

PROOF. For simplicity we just give an outline of this proof for two cameras. The proof for multiple cameras can be completed by induction.

Let p and $q \in \mathcal{C}_{\alpha_1,k_1} \cap \mathcal{C}_{\alpha_2,k_2}$ for some indices (α_i, k_i).
Part I:
First, consider cameras α_1 and α_2 on the same side of the line $L(p,q)$. We know that there exist convex monotone paths Γ_i connecting p to q such that $\Gamma_i([0,1]) \subset \mathcal{C}_{\alpha_i,k_i} \cap \Delta_{p,o_{\alpha_i},q}$ for $i = 1,2$ (see left plot in figure 11).

By lemma 5, we can choose a path Γ corresponding to a segment of the boundary of $\mathcal{R} := \mathcal{R}(\Gamma_1) \cap \mathcal{R}(\Gamma_2)$ (see right plot in figure 11). We choose the path that consists of segments from Γ_1 and Γ_2 so Γ will be feasible. We note that lemma 5 also tells us that Γ is monotone with respect to camera α_i (since $\mathcal{R}$ is between $L(p, o_{\alpha_i})$ and $L(q, o_{\alpha_i})$). Also,
$$\Gamma \subset \mathcal{R} = \mathcal{R}(\Gamma_1) \cap \mathcal{R}(\Gamma_2) \subset \mathcal{D}_{\alpha_1} \cap \mathcal{D}_{\alpha_2}$$
due to convexity of $\mathcal{D}_{\alpha_i}$.
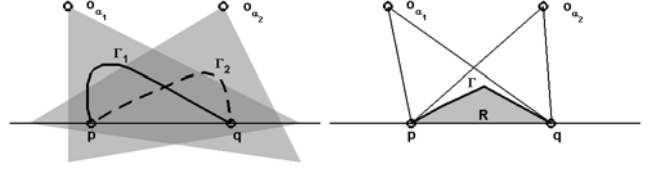

Figure 11: Illustration of the construction of Γ.

By lemma 8, we know that there are no objects inside the regions enclosed by $\overline{p o_{\alpha_i}} \cup \Gamma([0,1]) \cup \overline{q o_{\alpha_i}}$ (since otherwise there would be a bisecting line and we assumed that we already decomposed using all bisecting lines). Hence, $\overline{s o_{\alpha_i}}$ does not intersect any object for $s \in \Gamma([0,1])$, which implies that Γ is visible by both cameras (i.e. $\Gamma([0,1]) \subset \mathcal{C}_{\alpha_1,k_1} \cap \mathcal{C}_{\alpha_2,k_2}$).
Part II:
Now we consider cameras α_1 and α_2 at opposite sides of the line $L(p,q)$. There are two main cases to consider.
Case 1:
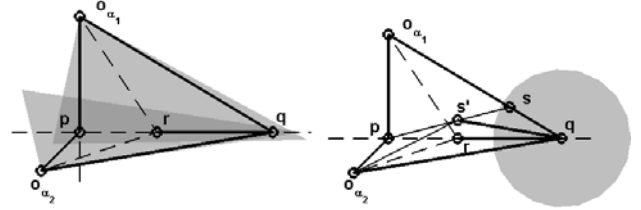For the first case we consider a configuration as seen in figure 12 (left).

Figure 12: Illustrations for Case 1.

We would like to conclude that the path $\overline{pq}$ is visible by both cameras. Assume that it is only visible up to a point r (not including this point since objects are closed). Then, an object must intersect $\overline{r o_{\alpha_1}}$ or $\overline{r o_{\alpha_2}}$. By lemma 8, we notice that the interior of $\Delta_{r,o_{\alpha_1},q}$ must be empty; otherwise, there would be a bisecting line.

Also, since $\mathcal{D}_{\alpha_2}$ is open, we can find a ball B around q that is contained in $\mathcal{D}_{\alpha_2}$ (see right plot in figure 12(right)). We choose a point $s \in B \cap \overline{q o_{\alpha_1}}$.
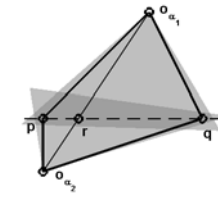
Then, $\Delta_{p,s,q} \subset \mathcal{D}_{\alpha_1} \cap \mathcal{D}_{\alpha_2}$. From there, we can choose a point s' as shown in the diagram such that $\overline{s'q} \subset \mathcal{D}_{\alpha_2}$ is feasible. This is possible since there are no objects in $\Delta_{r,o_{\alpha_1},q}$. However, if there was an object intersecting $\overline{r o_{\alpha_2}}$, then $L(r, o_{\alpha_2})$ would be a bisecting line. But, it is not. So, there are no objects intersecting $\overline{r o_{\alpha_2}}$.

Similarly, there are no objects intersecting $\overline{r o_{\alpha_1}}$. This means that r is visible by both cameras, which contradicts our initial assumption. This shows that $\overline{pq}$ is visible to both cameras, i.e. $\overline{pq} \subset \mathcal{C}_{\alpha_1,k_1} \cap \mathcal{C}_{\alpha_2,k_2}$.
Case 2:

For the second case, we consider a configuration as shown to the left.

By following the same analysis as before, we can show that $\overline{pr} - \{r\}$ and $\overline{rq} - \{r\}$ must be visible by both cameras. However, we could have an object in $\overline{o_{\alpha_1} o_{\alpha_2}}$. Nevertheless, objects must enclose some area which does not allow an object to be contained in this line. Therefore, $\overline{pq} \subset \mathcal{C}_{\alpha_1,k_1} \cap \mathcal{C}_{\alpha_2,k_2}$. $\square$