Semantic Localization for IoT

Matthew Weber and Edward A. Lee

Abstract Euclidean geometry and Newtonian time with floating point numbers are common computational models of the physical world. However, to achieve the kind of cyber-physical collaboration that arises in the IoT, such a literal representation of space and time may not be the best choice. In this chapter we survey location models from robotics, the internet, cyber-physical systems, and philosophy. The diversity in these models is justified by differing application demands and conceptualizations of space (spatial ontologies). To facilitate interoperability of spatial knowledge across representations, we propose a logical framework wherein a spatial ontology is defined as a model-theoretic structure. The logic language induced from a collection of such structures may be used to formally describe location in the IoT via semantic localization. Space-aware IoT services gain advantages for privacy and interoperability when they are designed for the most abstract spatial-ontologies as possible. We finish the chapter with definitions for open ontologies and logical inference.

1 Location as IoT Context

Today, we have mature theories of computation, developed over the last 80 years or so, and mature theories of physical structure and dynamics, developed over the last 300 years or so. But we have only the barest beginnings of theories that conjoin the two. One of the key points of friction is that the notion of location in space and time are central to a physical reality, but absent in a cyber reality. When the focus is mutual imitation, as in simulation, it is natural to construct cyber representations of space and time by approximating positions in a Euclidean geometry and Newtonian time with floating point numbers. But when the goal is the kind of cyber-physical

M. Weber (✉) · E. A. Lee
UC Berkeley, Berkeley, United States
e-mail: matt.weber@eecs.berkeley.edu

E. A. Lee
e-mail: eal@eecs.berkeley.edu

R. Pandey et al. (eds.), *Semantic IoT: Theory and Applications*, Studies in Computational Intelligence 941, https://doi.org/10.1007/978-3-030-64619-6_16

collaboration that arises in the Internet of Things (IoT), such a literal representation of space and time may not be the best choice.

When considering mobile devices and the IoT, applications often care more about logical spatial and temporal relationships than quantitative ones. To preserve security and privacy, for example, one device may be granted access to data held by another device only when the two devices are in the same room at the same time. The notion of "same room at the same time" is an example of what we call *semantic localization*. It is not so much about geometric location, but rather asserts a "semantic" spatial relationship.

In this chapter we survey IoT-relevant location models from robotics, the internet, cyber-physical systems (CPS)s, and philosophy. The diversity in these models is justified by differing application demands and conceptualizations of space (i.e. spatial ontologies). To facilitate interoperability of spatial knowledge across representations, we propose a logical framework wherein a spatial ontology is defined as a model theoretic structure. The logic language induced from a collection of such structures may be used to formally describe location in the IoT via semantic localization. Space aware IoT services gain advantages for privacy and interoperability when they are designed for the most abstract spatial-ontologies as possible. We finish the chapter with definitions for open ontologies and logical inference.

For all its importance to understanding IoT systems, localization, the challenge of determining the location of physical objects, remains an open problem. GPS, which has been a resounding success for outdoor localization, relies on direct line-of-sight signals from satellites, and is consequently ineffective for indoor environments or outdoor environments where obstructions, such as buildings, interfere with measurements. Researchers have been trying to address the indoor localization problem since the early 1990s with systems like Active Badges [1] and Cricket [2], and yet even to this day, a general purpose, accurate, cost effective, deployable system with the potential to reach the ubiquity of outdoor GPS remains elusive. A big part of what makes the problem difficult is the potential for interference in indoor environments where walls, furniture, and people, obstruct and reflect signals. Even something as simple as turning on a microwave oven causes interference to RF signals and might disrupt signal strength measurements for an indoor localization system operating in the 802.11 bands. Nevertheless, we are optimistic that in the near future, IoT applications will routinely have available a variety of types of location information with a range of quality. This chapter addresses how to organize and use that location information.

The most commonly articulated purpose for indoor positioning is indoor navigation. There is no doubt a market for apps that can help you find your way in whatever building you happen to be inside, but in our view this is probably a small market that dramatically understates the potential of contextual awareness in the IoT. The future of indoor and outdoor space-aware IoT systems involves scenarios where position in space is less important than spatial interrelationships. Consider a fleet of self-driving cars, where proximity in driving time, energy, and ride sharing opportunities are more useful criteria for control than geo-coordinates. Indoors, having awareness of which devices are in the same room may be more useful than measurements of their position in two or three-dimensional space. For such applications, different representations of space than a coordinate-system based physical map become appealing.

Relational ontologies of space aren't wildly foreign concepts; they can be found in some of today's apps. Take data from FourSquare, the app that lets users "check into" locations, as an example of a non-geometric representation of space. A user checked into a restaurant on FourSquare is known to be inside the establishment, but it would be a mistake to guess exact geocoordinates for him/her and plot them inside the restaurant's perimeter because they might be sitting at a table or standing by the door, and precise geocoordinates would suggest a false confidence as to the nature of unknown information. Unplotability doesn't make the FourSquare data somehow less accurate or reliable than a physical coordinate map, it just makes it different. We call this kind of *geometrically fuzzy yet logically precise spatial information* **semantic localization**.

1.1 Designing a Robo-Cafe

In collaboration with researchers at U Penn, Michigan, UW, CMU, and Berkeley, in 2015 we demonstrated a robotic delivery system at the DARPA "Wait, What?" conference where users could place an order on a smart phone localized by the ALPS Ultrasound Localization System [3] and have a desired snack delivered to their location by a roaming Scarab Robot [4]. The demo was designed to showcase integration and composability of IoT systems via accessors [5], but most relevant to this chapter is the spatial interaction needed between the Scarab and ALPS.

The Scarab comes equipped with a laser rangefinder which it uses with standard ROS packages to perform Simultaneous Localization and Mapping (SLAM) and to build an occupancy-grid map of its environment (see Fig. 1). An occupancy grid is a fairly simple data structure commonly used in robotics to represent an environment (modeled as a grid over 2D or 3D Euclidean space) that is essentially a big array with values from 0 to 100. A value of 0 indicates the robot is almost certain the cell does not contain an obstacle, and a value of 100 that the cell is almost certainly impassable. The robot also maintains an estimate of its pose (position and orientation) at the cell where it is currently located.

The second localization system, ALPS, uses ultrasonic beacons, and is also deployed in the DOP Center (Fig. 1). The system is deployed by placing beacons at known locations in a building and finding the correspondence between the beacons and coordinates on the building's floor plan. The beacons send time synchronized chirps of ultrasound in the 20–22 kHz bands that are beyond the range of human hearing but receivable at the standard sampling rate of a cell phone microphone. A smartphone with an ALPS app can locate itself on the floor plan's coordinate system.

When the robot is localized on its occupancy grid and the phone is localized on the floor plan, the Scarab uses ROS navigation packages to deliver a snack. However, there is a rather subtle challenge in the last step: the phone has known coordinates on the floor plan and the robot is at a known cell of the occupancy grid, but the two are, as

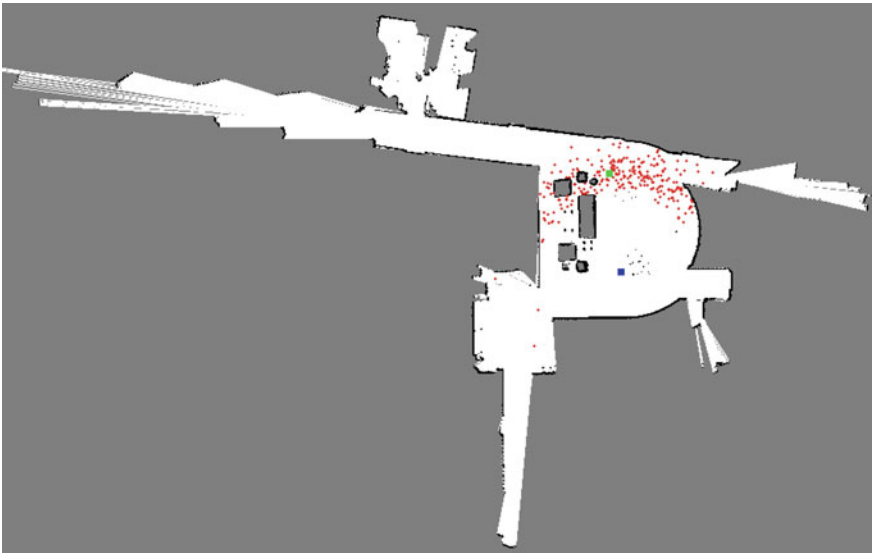

Fig. 1 Occupancy grid formed by a Scarab robot roving the DOP Center at Berkeley. This image shows use of a relatively poor distance sensor on the roving robot, measuring for example received signal strength from another object, and then applying a particle filtering algorithm constrained by the occupancy grid map to estimate the position of the other object. The red dots are the particles, the green square is the target, the blue square is the Scarab robot, and the black areas are occupied grid points as detected by the lidar rangefinder on the Scarab. The grey areas indicate where the occupancy grid has no information. Image courtesy of Ilge Akkaya

given, totally unrelated! Deployment of the Robo-cafe requires a coordinate system alignment phase in which ALPS's model of space is brought into concordance with the Scarab's model.

The coordinate system alignment problem in Robo-Cafe is in fact an instance of a general problem that must be addressed whenever two IoT systems seek to work across contextual ontologies. Usually when IoT systems are designed by different engineers working with different conceptualizations of space, spatial information cannot be shared between systems without additional translation. A central motivation for the modeling framework presented in this chapter is to formalize the structure of spatial ontologies for the development of mappings and relationships that enable heterogeneous mixtures of ontologies in IoT applications. We discuss a formalism for such cross-ontology reasoning in Sect. 2.1.

1.2 Spatial Ontologies

Location is one of the most important and challenging aspects of physical context. Location matters for the IoT in ways it does not for the Internet. There's a world of difference between illuminating a smart light bulb located in your home or one a thousand miles away. But while the physical location of a web server might affect the latency of communication or quality of service, it won't fundamentally change the content of the hosted page. For an IoT device, its physical relationship with the world has everything to do with what it can and cannot accomplish.

Any IoT system that seeks to interact with the physical world assumes a model of space, either explicitly or implicitly. Such a model is a spatial ontology. Broadly, the subject of ontology from philosophy is a study of the nature of existence, what it means for something to be and to be something. In computer science, ontology is usually about association of entities in a model with structured taxonomies, addressing questions like "is this object an instance or example of that class of objects?" (taxonomy) or "is this object *a part of* an instance or example of that class of objects?" (meronymy) relationships. In prior work, it has been shown that useful ontologies can be constrained to have a mathematical lattice structure, and that they thereby acquire enormous algorithmic and formal benefits that can be leveraged to compose ontologies, perform inference, and check correctness [6–8]. Such ontologies form a subset of commonly used ontology frameworks such as Web Ontology Language (OWL). Their mathematical structure resembles that of Hindley-Milner type systems, from which they inherit practical algorithms that scale to very large numbers of elements. For example, type inference maps into the problem of finding a fixed point of a monotonic function over a lattice.

Spatial ontologies have more diversity than just choice of coordinate system. A common dichotomy in ontologies is the distinction between "objects" and "fields" [9–11]. An "object" is an entity that is distinct, with a clear boundary, and in the language of [9] is "individual and fully deniable." Examples of objects include: an apple, a table, or a flashlight. A "field" describes phenomena without clearly defined boundaries that are "smooth, continuous and spatially varying" [9]. The magnetic field emanating from a hand-held bar magnet is a good example of this concept. From a certain pedantic perspective, the field is present everywhere in the universe, only its strength is almost everywhere so weak as to be negligible. Some geographical features like lakes have elements of both objects and fields because it can be hard to identify where they end.

Spatial ontologies can also vary with respect to their interpretation of entities with respect to time. SNAP and SPAN are two cooperative ontologies proposed by Grenon and Smith [10] to capture the distinction between "continuants," objects with an identity that persists across time, and "occurants," processes defined in part by their beginning and ending. Examples of continuants include the planet earth or a pair of shoes because it makes sense to consider their spatial properties at a particular snapshot of time. The same is not true for occurants like a volcanic eruption or the takeoff of a helicopter. Such occurants unquestionably have a spatial existence but their reality is best comprehended in four full dimensions; a sequence of 3D observations misses something essential about the nature of the process. There is clearly a strong interrelation between SNAP and SPAN ontologies. This point is not missed by Grenon and Smith, who devote a latter section of their paper to trans-ontology interrelations between SNAP and SPAN.

1.3 Semantic Technologies

The term "semantic technology" describes a collection of popular standards and technologies for representing and working with ontologies, be they spatial or otherwise.

The Semantic Web was proposed by Tim Berners-Lee in 2001 as an extension of the World Wide Web that would allow ordinary HTML web pages to be enhanced with special markup to label their semantic content. The hope is that when markup is combined with a collection of ontologies for web content and real-world objects, algorithms will be able to apply ontological reasoning to web elements and data. For example, an image of a bridge embedded in a web site could be labeled as such and found through a general search for "landmarks" by using the ontological information that a bridge *is* a landmark. According to the wikipedia article on the semantic web, by 2013 some 4 million web pages had been augmented with semantic web information. But this is done primarily through human intervention, which could account for the relatively modest penetration compared to the total number of web pages.

Semantic Web ontologies are expressed in Resource Description Framework (RDF), an abstract model for semantic data as sentence-like statements about the world in triples of subject, predicate, object. For example: the sentence "A cow" (subject) "isa" (predicate) "farm animal" (object), or "The mall parking lot" (subject) "has the number of free spaces" (predicate) "45" (object). As hinted at by these examples, triples can express both abstract information about classes (cows and farm animals) as well as facts about specific instances (the mall parking lot) and raw data values (45). A database designed and optimized for RDF data is known as a semantic repository or alternatively a triple store. The W3C SPARQL Protocol and RDF Query Language (SPARQL) recommendation [12] defines both a protocol and a query language for performing SQL-like operations on a semantic repository such as queries, inserts, updates, and deletes.

RDF is a natural way to express relational ontologies, as discussed in Sect. 2.3, for semantic localization. Additionally, some semantic repositories, like GraphDB, support geospatial plugins for efficient queries over geocoordinates (i.e. latitude and longitude pairs). If compatible with the GeoSPARQL standard [13], the semantic repository may also be able to automatically derive RCC (Region Connection Calculus) relationships, such as containment of one geospatial object within another, directly from the definitions of the objects themselves.

1.4 Standards for Spatial Representation

Many standards for spatial representation have been proposed in different domains, a sample of which is presented here.

According to Lieberman et al., as of 2007 the semantic web maintained at least seven varieties of spatial ontologies [14]. These include Geospatial Features, Feature

Types, Toponyms/Placenames, (Geo) Spatial Relationships, Coordinate Reference Systems, Geospatial Metadata, and (Geo) Web Services. The relationships between these, however, are highly unstructured and lacking in formal properties that can exploited algorithmically. More recently, geospatial ontologies like GeoDataOnt [15] have been developed to provide a unified ontology for this domain.

A popular (non-RDF) spatial ontology today is codified in a JSON schema called GeoJSON [16]. This is used by many location based services. In contrast to the semantic web, GeoJSON is good at representing geometries, but not higher level ontological concepts and relationships. It supports points, lines, polygons, and collections of polygons in 2D or 3D. Given the extensive support for GeoJSON in existing apps and software, it is a useful standard to leverage for geometric concepts. But restricting spatial ontologies to exclusively geometric concepts is a mistake. Spatial relationships are more complex.

On the opposite end of the complexity spectrum, the Open GIS Geography Markup Language (GML) Encoding Standard [17] is a 437 page specification document describing an XML schema for spatio-temporal ontologies. It follows the ISO 19101 definition of a feature as an "abstraction for real world phenomena" and represents the world as a collection of features defined as name, type, value triples. The increased complexity allows for the description of more sophisticated data such as spatial geometries, spatial topologies, time, coverages, and observations. The format can be extended to application schema such as IndoorGML [18] which is targeted for indoor navigation. IndoorGML focuses on layered graph representations of relationships such as adjacency and paths between semantic objects in indoor space. It models the world as a collection of cells representing geometry and topology via the Poincaré duality to achieve a "Multi-Layered Representation" of a given space in different contexts.

A variety of geometric data structures and algorithms are employed in the field of computational geometry when high performance is desired for computationally difficult spatial analysis [19]. For example, a doubly-connected edge list is used for the thematic map overlay problem, in which the overlay of spatial subdivisions is computed.[1] A trapezoidal map is another geometric data structure employed to solve point location queries: given the coordinates of a point and a map subdividing the plane into regions, determine which region contains the point.

Point clouds are another computationally useful format for spatial information in the domain of computer vision. Visually oriented sensors such as stereo cameras or time of flight cameras (e.g. Light Detection and Ranging, LiDAR) measure the location of individual 3-dimensional points in the world. These points represent sampled measurements of real-world objects. Once collected, software such as the open source point cloud library [20] can use a point cloud data set to reconstruct a sampled surface or perform segmentation to semantically identify objects.

The diversity of these standards for spatial representation is daunting. Yet it is easy to see how applications in the IoT with different purposes for spatial information and different sensors for collecting that data benefit from different data representations.

[1] Imagine overlaying two circles to form a venn-diagram, but with polygons instead of circles.

Engineers of spatial IoT systems are tasked with finding the best location model for their application. But IoT systems designed for interaction with other data sources must additionally be created with an understanding of how a selected location model is related to the location models of other sensors and IoT systems in the environment.

2 Location Modeling

The purpose of location modeling in this work is to support a logic for reasoning about spatial ontologies across independently designed IoT systems. By moving beyond just geometric position, this logic offers the possibility for a much richer set of applications than just navigation, including for example security (e.g. restricting access to some service to only devices in the same room); asset tracking (e.g. where is the remote control for this device, or the device for this remote); spatial search (e.g. find a temperature sensor in the same room as a mobile device); commissioning (e.g. deploying sensors and actuators without manually specifying their location); and context-aware services (e.g. lighting systems that automatically adjust to usage patterns of a room). We believe semantic repositories are a good start for this, but there is room for a larger suite of software components and services for creative application designers to use when reasoning about location information.

Such services could handle mobility (e.g. notification when a device is no longer in the same room) and superposition of disjoint maps constructed at different semantic and geometric layers (e.g., relating geometric information to "in the same room" semantic information).

2.1 Model Theory

Model theory is a domain of mathematical logic originally developed to analyze logical formulas regarding mathematical structures such as groups, graphs, and fields. The key observation behind model theory is that logical formulas can be written to express properties in a manner independent of the mathematical structures with respect to which they are evaluated. For example the formula $\exists n\ 0 < n < 1$ is true with respect to $\mathbb{R}$ or $\mathbb{Q}$, but not with respect to $\mathbb{Z}$ or $\mathbb{N}$. A model (or structure) specifies a domain, such as $\mathbb{R}$, and gives interpretations to the symbols 0, 1, and $<$ so that their particular relationship may be determined. We summarize the fundamentals of model theory relevant to CPS location modeling below. The main reference for the following definitions is [21], which may be referred to for a more comprehensive introduction to model theory.[2]

A **formula** is a logical statement constructed in the usual way from:

[2] A friendlier introduction can be found at https://plato.stanford.edu/entries/modeltheory-fo/

- logical symbols: ($\rightarrow$, $\leftrightarrow$, $\neg$, $\wedge$, $\vee$, $\forall$, $\exists$,), and
- variables (a countably infinite collection)
- function symbols (e.g. $+$ for a group operation)
- relation symbols (e.g. $\leq$ for the ordering relation on R)
- the relation symbol $=$, as the usual "equal sign"
- constant symbols which represent a particular element from the domain (e.g. 0 or π).

The arity of function and relation symbols is ≥ 1.

A **signature** is a particular set of function, relation, and constant symbols. The **language** of a signature is the set of well-formed formula expressible using functions, relations and constants from the signature. A variable v_0 is **bound** iff it appears in a subformula (i.e. a syntactically correct part of a formula) following ($\forall v_0$) or ($\exists v_0$). Otherwise the variable is **free**, and may be assigned a value separately. For example: formula ϕ with free variables $v_0, v_1, \ldots, v_k$ may be written as $\phi(a_0, a_1, \ldots, a_k)$ to express the assignment of a_0 to v_0, a_1 to v_1, and so on.

The **sentences** of a language are formulas of the language with no unbound variables. A **structure** (or model) is a tuple $\mathbb{A} = \langle \mathbf{A}, I \rangle$ where **A** is a domain, i.e. a non-empty set, and I is an interpretation function. I maps function, relation, and constant symbols to functions defined over **A**, relations defined over **A**, and elements of **A** respectively. A structure $\mathbb{A}$ **models** a sentence S of a language when the interpretation of the sentence within the structure evaluates to true. This relationship is denoted by $\mathbb{A} \vDash S$ and its converse by $\mathbb{A} \nvDash S$.

A language with a finite signature may be concisely written for example as, $\mathbb{L} = \{< , 0, 1\}$. Similarly, a model's domain and interpretation for that finite signature may be informally written as an analogous tuple, e.g. $\mathbb{A} = \langle\, \mathbb{R}, \mathbf{<}, \mathbf{0}, \mathbf{1} \rangle$. Here, $\mathbb{A}$ is the structure with domain $\mathbb{R}$ which interprets $\mathbb{L}$ with the strict ordering relation $<$, and constants 0 and 1.

Putting it all together, we may now formalize the motivating observation from the beginning of this section that the same formula may be true or false with respect to different domains. Regarding the example formula $\exists n\ 0 < n < 1$ we have $\mathbb{A} \vDash \exists n\ 0 < n < 1$, but for $\mathbb{B} = \langle\, \mathbb{N}, \mathbf{<}, \mathbf{0}, \mathbf{1} \rangle$, $\mathbb{B} \nvDash \exists n\ 0 < n < 1$.

2.2 Semantic Localization

We propose using the concepts of model theory to formally describe location in IoT systems. A spatial ontology can be represented as a structure $\mathbb{A} = \langle \mathbf{A}, I \rangle$. For $\mathbb{A}$ to be useful as a model of the space, most likely the elements of **A** should be places or things located at places. Similarly, I should provide spatially meaningful interpretations of relations, functions, and constants. The language for such a structure will then consist of semantic localization statements.

Defining semantic localization as a model-theoretic language has the advantage of separating the specification of spatial reasoning from its implementation within a particular spatial ontology. Just as the formula $\exists n\ 0 < n < 1$ may be evaluated within different structures, so too might a semantic localization formula be evaluated within heterogeneous spatial ontologies. For example, let *contains* (*a, b*) be a binary relation which is true when room *a* contains person *b*, let *user*1 and *user*2 be constants for people, and let the variable *room* range over a set of rooms. The formula $\exists$ *room contains*(*room, user*1) $\wedge$ *contains*(*room, user*2) expresses the spatial arrangement in which *user*1 and *user*2 are both within the same room, independently of a particular spatial ontology. We propose using semantic localization as a conceptual interface between location programming and location models in the IoT.

A semantic localization formula can be interpreted in one of two ways: either as an event condition or as a query into some spatial database. In the first case, the sentence acts as a predicate that triggers an event when it evaluates to true. In the second case, the formula can be evaluated against database entries to signify that the entries to return are those that cause the formula to evaluate to true when plugged into unbound variables. However, in either case a statement can only be evaluated in an ontology with a compatible signature.

Figure 2 represents a central idea governing location modeling, relating mathematical structures to the spatial connectives (relations) of physical objects in a CPS which they are capable of evaluating. Applying the concepts from model theory to CPS location modeling has the added advantage of enabling mathematical analysis to bring the theorems of model theory to bear on the relationships spatial ontologies have to one another.

As suggested in Fig. 2, spatial ontologies may be used to reason about spatial connectives, or spatial connectives may be discovered by sensors and used to construct mathematical structures. Relations represent the structural aspects of the

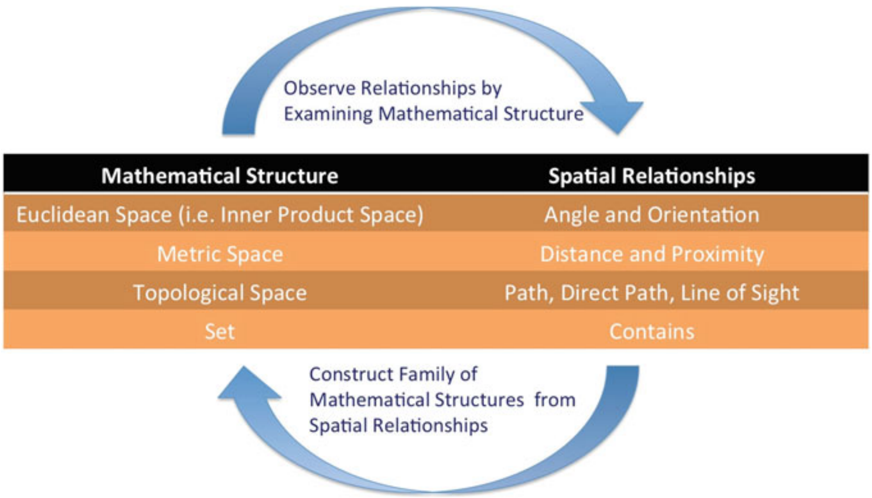

Fig. 2 A comparison between mathematical structures and corresponding evaluable spatial relationships as described in our previous work [32]

space (e.g. containment, path, proximity, angle, etc.), and functions define other structural aspects of the space (such as distance for a metric space) as appropriate. The values of constants, relations and functions are potentially time varying as the structure evolves. For instance, a topological ontology of an indoor space with doors opening and closing has a dynamic “path” relation. The quality and nature of the sensor data may constrain the level at which these ontologies may be constructed. For example, orientation information may simply not be available.

2.3 Physical and Relational Ontologies

In the previous section, a spatial ontology is a mathematical structure which can be used to evaluate a logical sentence that makes reference to spatial relationships. This notion of a spatial ontology is considerably more general than the usual notion of a printed paper map with a 2D representation of the road network of a city, for example. We will use the term “relational ontology” when we want to emphasize the abstracted nature of the spatial relationships that the map represents, but in this research, a spatial ontology is a mathematical object at any of these levels of abstraction, as long as it encodes some form of spatial relationships. For example, Fig. 3 shows a relational ontology that is a partial order induced by the containment relation between sets; the relational ontology does not say anything at all about geometric properties such as distance or orientation.

We have arrived at an important principle: *Space-aware services should be constructed for the signatures of the most abstract spatial ontologies as possible.* This will enable them to operate in more sensor-poor environments, to benefit from a greater variety of sources of spatial information, and to better preserve privacy by

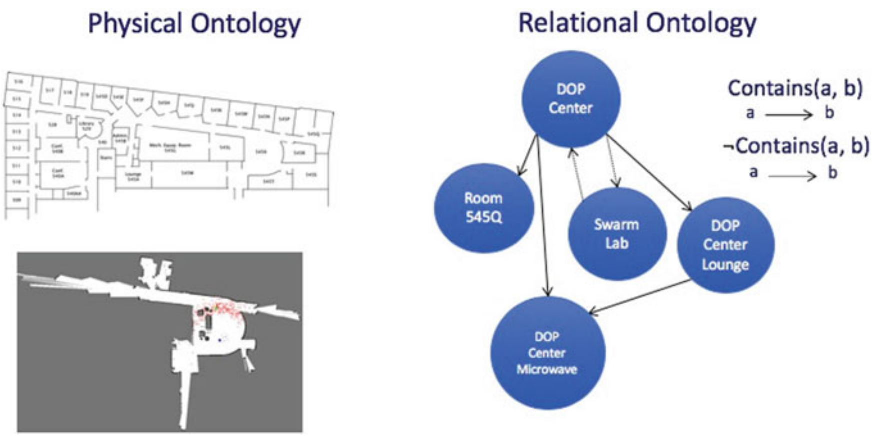

Fig. 3 Concrete examples of Euclidean-space ontologies vs. an abstracted relational ontology that represents only containment relations

not handling information that is not needed. For example, the set containment relation is all the map information necessary for the FourSquare localization example in the introduction, since the only information to be gained from checking in is containment.

Consider the advantages of applying this maximum abstraction principle to the spatial ontologies depicted in Fig. 3. All three ontologies, the occupancy grid, the floor plan, and the abstract graph, represent information about the same region of space. An IoT application could theoretically use any one of the ontologies to determine, for example, that room 545Q is inside the DOP Center. However, it takes a certain level of geometric understanding to extract that information from the less abstract physical ontologies. To use the occupancy grid, our IoT application must be equipped with an algorithm for parsing occupancy grids and determining when a collection of cells in a grid is contained by another collection of cells. In other words, effective use of the occupancy grid is restricted to IoT applications that are prepared in advance to interact with robotic maps. Similar limitations hold for IoT systems using the floor plan, or any other spatial ontology requiring geometric analysis.

But if the IoT system were designed to interact with map providers through an abstract notion of containment, the system wouldn't have to bother understanding the nuances of geometry in every spatial ontology it might come across. It may instead operate in terms of semantic localization. Perhaps the relational ontology was created by inspecting an occupancy grid, or maybe it was a floor plan. Either way the IoT application doesn't have to bother knowing the specifics. As long as it can pose the query regarding the DOP Center, room 545Q, and containment, the IoT application can treat the source of an abstracted spatial representation as a black box.

As they get more abstract, of course, relational ontologies lose the ability to evaluate some kinds of spatial relationships. This idea parallels the usual hierarchy of mathematical spaces. A Euclidean space has quite a lot of mathematical structure that may not match well with the information available sensors are able to deliver. A Euclidean-space ontology supports reasoning about angles and orientation, concepts that are not defined in the more abstract mathematical structures shown in Fig. 2.

The hierarchy of these mathematical spaces offers a starting point for reasoning about combinations of maps. For example, given a Euclidean-space map of an office space and a Set (containment) map of objects in the space, objects can be placed approximately, with known error bounds, onto the Euclidean-space map. But much more complicated mapping combinations will be required, since even two Euclidean-space maps may not use the same coordinate system. The concept of a spatial ontology becomes an essential feature of location modeling.

Topological spaces can be used to construct maps that represent paths through indoor settings. Navigation with graphs is a common concept in robotics [22], where nodes represent waypoints in a space and edges represent paths between waypoints. Such data structures are routinely used to construct sequences of actions to move a robot between nodes. Additionally, Ghrist et al. [23] show that algebraic topology can be used directly to relate the convex hull of a landmark set in a Euclidean space to a simplex of a simplicial complex. This provides a natural abstraction mechanism for topological maps.

Non-Euclidean metric maps are useful when the standard Euclidean metric does not really capture the interesting properties of a space. Consider a point x on the third floor of a building and the point y directly below it on the second floor. Points x and y are very close to each other in Euclidean space, but for the purposes of navigation, this misrepresents reality. We can instead define a metric space with metric D, where

$$D(x, y) = \begin{cases} \text{minimum length of a continuous path from x to y,} & \text{if there is such a path} \\ \infty, & \text{otherwise} \end{cases}$$

This is easily shown to be a metric (or even an ultrametric, for some graph-structured metric spaces). If stairways and elevators are not navigable open space for a particular robot, then this metric will yield $D(x, y) = \infty$, considerably more than the Euclidean distance.

An inner product space (of which a Euclidean space is a common variety) introduces the notion of angles. Angles can facilitate special kinds of analysis like trilateration, and the use of trigonometric angle measurements to localize objects in coordinate space.

As these examples illustrate, there are practical reasons to construct non-Euclidean ontologies. However each of these mathematical ontologies has the property that any map entity placed at a particular coordinate takes on all spatial relationships to other map coordinates implied by the structure of the space. This is undesirable when only a portion of those relations are positively known to be true and the rest are unknown. A key advantage of relational ontologies is the expression of *open ontologies*, where the absence of a relation does not imply its converse. This is analogous to ancient maps that provided useful navigation information despite significant distortions in the geometry and large gaps labeled "*terra incognita*". Open ontologies translate naturally into action plans that can deal with incomplete information.

This increased flexibility comes at the cost of a slightly more verbose vocabulary for relations. Consider the containment map on the right hand side of Fig. 3. Because this ontology is open, knowing that one place is not contained by another isn't enough to know they have no space in common. Another relation, "disjoint," is necessary to express that positive fact explicitly. We hope the reader can see a connection here to intuitionisitic logic, in that for open ontologies it is not enough to know a spatial relation is not not true to infer that it is true. Instead, relations must be constructively built up from known facts.

2.4 Formalizing Open Ontologies

An open ontology $\mathbb{A}$ is a way of expressing partial knowledge about a spatial structure. If we take the philosophical position that the unexpressed information in an open ontology is fundamentally unknowabable, there is nothing to be done to increase the amount of information represented in $\mathbb{A}$. However, if we assume the missing information is knowable and could be expressed in an idealized (but hypothetical[3]) ontology $\mathbb{A}^*$, we may consider logical inference as a means to obtain information available in $\mathbb{A}^*$ but not $\mathbb{A}$.

We formalize this notion below, but first some definitions. Let $\perp$ be the symbol for "unknown".[4]

Definition 1 (Partial Order on Functions)
We define a (pointwise) partial order on n-valued function f: $\mathbf{A}^n \rightarrow (\mathbf{A} \cup \perp)$ with $f \leq f'$ iff for $x \in \mathbf{A}, f(a_1, a_2, \ldots a_n) = x \rightarrow f'(a_1, a_2, \ldots a_n) = x$.

Observe this definition allows $f(a_1, a_2, \ldots a_n) = \perp$ with $f'(a_1, a_2, \ldots a_n) = x$. In other words, f agrees with f' everywhere where f is not unknown, but may disagree where f is unknown.

Let open ontology $\mathbb{A}$ and its idealized $\mathbb{A}^*$ both be structures with the same signature and the same domain. $\mathbb{A}$ may be missing some information available in $\mathbb{A}^*$.

Definition 2 (Partial Order on Open Ontologies)
We define a pointwise partial order on open ontologies $\mathbb{A}$ and $\mathbb{A}^*$ with the same signature and domain ($\mathbf{A}$) by ordering relation $\sqsubseteq$. The $\sqsubseteq$ relation indicates $\mathbb{A}^*$ has more information than $\mathbb{A}$ when:

- $\mathbb{A}$'s functions may have unknown value ($\perp$) over some elements of the domain where $\mathbb{A}^*$'s functions are known. With f_{A} as the interpretation of function symbol f in $\mathbb{A}$ and $f_{\mathbb{A}^*}$ as the interpretation of f in $\mathbb{A}^*, f_{\mathrm{A}} \leq f_{\mathbb{A}^*}$.
- $\mathbb{A}$'s relations may be missing tuples which are available in the analogous relations of $\mathbb{A}^*$. For example, with r_{A} and $r_{\mathbb{A}^*}$ as interpretations of relation symbol r in models $\mathbb{A}$ and $\mathbb{A}^*$ respectively, $r_{\mathrm{A}} \subseteq r_{\mathbb{A}^*}$.
- $\mathbb{A}$'s interpretation of constant symbols may be less complete than the interpretation of $\mathbb{A}^*$. With k as the set of constant symbols in $\mathbb{A}$'s signature and c: $k \rightarrow (\mathbf{A} \cup \{\perp\})$, as the function mapping constant symbols to domain elements, $c_{\mathrm{A}} \leq c_{\mathbb{A}^*}$.

Not only does the ordering relation defined by $\sqsubseteq$ relate $\mathbb{A}$ to $\mathbb{A}^*$, it also relates $\mathbb{A}$ to a chain of non-idealized open ontologies $\mathbb{A} \sqsubseteq \mathbb{A}' \sqsubseteq \mathbb{A}'' \sqsubseteq \ldots \sqsubseteq \mathbb{A}^*$ with progressively more information than $\mathbb{A}$. Applying a logical inference procedure to $\mathbb{A}$, and filling in an unknown function, relation, or constant with a concrete value can be interpreted as finding an $\mathbb{A}'$ with $\mathbb{A} \sqsubseteq \mathbb{A}'$.

It may not be possible to definitively determine whether or not an open ontology models a formula which depends on unknown functions, relations, and constants. If the true/false value of a formula depends on evaluating a function where it is unknown, an unknown constant, or the negation of a relation which is not explicitly given in the model, the formula may not be evaluated with respect to the open model.

[3]Of course we don't actually know the contents of A* because it contains the information we currently don't know in A. But it is nevertheless useful to define A* as a model so we may make explicit our assumptions about the missing information.

[4]We do not always explicitly augment the domain of an open ontology to include $\perp$, but this may be assumed.

The advantage of an open ontology is the ability to evaluate formula regarding *known* information without being forced to make questionable assumptions on the unknown parts of the model.

The next section provides some examples of valid logical inference procedures for open relational ontologies.

3 Logical Inference on Ontologies

Consider the relational ontology on the left hand side of Fig. 4. Nodes in the map represent objects or places in the world, and dark edges signify a known upper bound on the distance between them in some metric given by the weight of the edge. Since this is an open map, the absence of a black edge does not signify the converse of proximity (which we might call "anti-proximity"); if we want to express anti-proximity in this graph we must explicitly designate it with a dashed line edge.

This being a metric space, we can apply the triangle inequality to the graph and note that if A and B are within 30 meters and if B and C are within 30 meters, then A and C must be within 60 meters. Before we add this edge to the graph as shown in the right hand side of Fig. 4, we may note that the triangle inequality applied to the edge from A to D and from D to C gives a tighter bound and express that A and C must in fact be within 40 meters of each other.

Next consider the example in Fig. 5 with an anti-proximity edge drawn from A to C. This indicates that A and C are known to be *at least* 10 m apart, whereas the proximity edge indicates that they are *at most* 40 m apart. Applying the contrapositive of the triangle inequality gives a relational ontology in which at least one of A or C must be more than 5 meters away from another node E. This matches the intuitive notion that for two objects known to be far away from each other; at least one of them must be somewhat distant from any third object. Note that this data structure is more than a simple graph now, since there is appended a disjunction between the two edges to E.

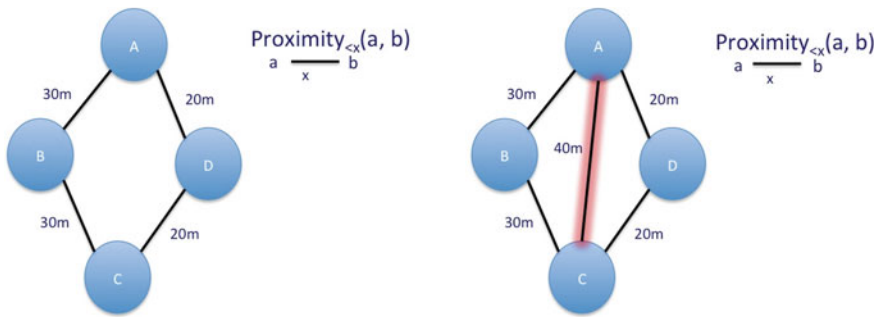

Fig. 4 An example of logical inference for a relational proximity map

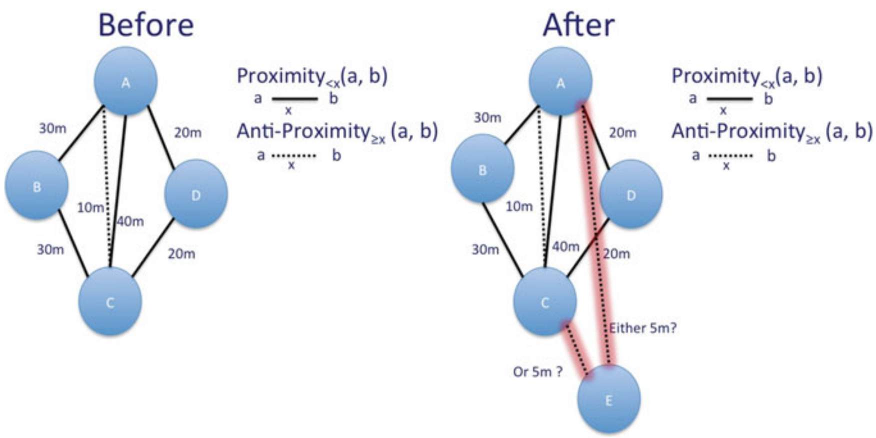

Fig. 5 Another example of logical inference for anti-proximity

A relational ontology of distance for a Euclidean space permits more sophisticated inference methods. A considerable amount of research has been undertaken in the sensor network community to find a Euclidean space embedding for a weighted undirected graph such that the Euclidean distances between nodes in the embedding match the edge weights in the graph. If such an embedding is successfully found, it is possible to infer internode distances not explicitly specified. One such algorithm [24] uses a process of iterative trilateration with robust quadrilaterals where three nodes with known Euclidean position (say A, B, and C) are used to establish the position of a connected node (D). Once the position of D is established, it can be used in the next iteration of the algorithm as a reference point to give the position of some other node E.

In addition to determining unknown inter-node distances, the properties of a Euclidean space also facilitate detection of inconsistent edges signifying outlier measurements. In prior work, [25] we expanded upon an algorithm given in [26] which uses graph rigidity theory to identify components of a graph that admit only a specific embedding. If a questionable edge is wildly inaccurate, it can be identified by considering other rigid subgraphs that are consistent with Euclidean geometry.

These sorts of Qualitative Spatial Reasoning (QSR) received significant research attention in the 1990s. The main focus of this work was the construction of formal algebras for inference on qualitative spatial relationships. For example, Frank's calculus for cardinal directions and informal distances such as "near" and "far" can infer such relationships for unknown cities given knowledge on how they are related to a known city network [27]. Arguably, the most notable outcome of QSR today is the Region Connection Calculus (RCC) for 2-dimensional mereology (the part whole relationship) and topology [28]. RCC laid the foundation for the GeoSPARQL standard [13], which is today widely (but incompletely) implemented by modern semantic repositories[5] to leverage RCC relationships for queries on geospatial data sets.

4 Related Formal Structures from AI and Robotics

Pereira's BigActor model [29] gives a formalism for mapping with many similarities to our approach. Specifically he defines two kinds of spatial structures: a logical-space model with a rough correspondence to what we would call a relational ontology, and a physical-space model corresponding to a coordinate-based physical map. Pereira requires the same relations hold true between the same objects in physical and logical space. The model-theoretic proposal for semantic localization in this chapter can be seen as a generalization of Pereira's approach to include more diverse kinds of spatial structures.

Similar ideas to relational ontologies have been around in the world of AI and robotics research for some time [22, 30, 31]. However, where semantic localization is designed to integrate modeling and programming for heterogenous IoT systems, the focus of research in this domain is commonly inference and autonomous decision making. As an additional point of contrast, spatial modeling in robotics is usually from the perspective of a robot as it moves from place to place, but spatial modeling for localization systems is usually from the perspective of a place as people (or robots) move within.[6]

The distinction between absolute and relative space is raised by Vieu [31]. The elements of a spatial ontology are Basic Entities (is the space composed of points or basic regions?), Primitive Notions (topology: relating to contact and part-whole relationships; orientation: absolute, intrinsic, and contextual; distance: metric functions and discrete distance notions), and bounded/unboundedness. Vieu goes on to overview actual approaches researchers have used to represent space. Vieu also examines the difference between 3D space composed with time and 4D views.

Kuipers [22], in a classic robotics paper, introduces an ontology for spatial information flow from sensor values to, ultimately, 2-D geometry. His ontology allows information to be incomplete at different levels. For example the graph-topological connections between different maps may be known even if each of the maps hasn't been entirely fleshed out.

An example of the advantages of combining physical maps with relational information for robotic localization was demonstrated by Atanasov et al. [30]. The authors use set-based identification of semantically interesting indoor objects such as chairs

[5]Essentially a semantic repository is a database for relational data.

[6]We refer to models of things moving through space as "Lagrangian Models" and models of space with things moving within as "Eulerian Models". The terminology comes from the analysis of fluid flows.

and doorways to localize their robot, instead of the more commonly used techniques that use edges and corners in the field of view without consideration for their semantics.

5 Conclusion

In this chapter we observe that spatial models used in IoT applications frequently have good reason to be domain specific. We propose semantic localization as a unifying interface between spatial modeling and spatial programming. This abstract approach is motivated by the need to reconcile diverse spatial representations for cross-domain interaction. By treating spatial models as mathematical structures from model theory, the language of mathematical logic becomes an effective tool for describing the qualitative spatial relationships important for developing contextually aware IoT services. When space aware services are designed for abstract spatial ontologies, they gain advantages in privacy and interoperability.

Semantic localization focuses our discussion of physical and relational ontologies in which information may be expressed through mathematical coordinates, spatial relationships, and non-Euclidian maps of an environment. We formalize the notion of an open ontology with partially unknown information, and give examples of logical inference on open ontologies. Open relational ontologies are promising for developing contextually aware IoT services, and have a conceptual match with semantic web technologies such as RDF, SPARQL, and semantic repositories. Semantic localization gives a principled foundation for location modeling and the design of spatially aware IoT systems.

Acknowledgements The work in this chapter was supported in part by the National Science Foundation (NSF), award #CNS-1836601 (Reconciling Safety with the Internet) and the iCyPhy Research Center (Industrial Cyber-Physical Systems), supported by Camozzi Industries, Denso, Siemens, and Toyota.

References

1. Want, R., et al.: The active badge location system. ACM Trans. Inf. Syst.(TOIS) **10**(1), 91–102 (1992)
2. Priyantha, N.B., Chakraborty, A., Balakrishnan, H.: The cricket location-support system. In: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, pp. 32–43 (2000)
3. Lazik, P., et al.: ALPS: a bluetooth and ultrasound platform for mapping and localization. In: ACM Press, pp. 73–84 (2015)
4. Michael, N., Fink, J., Kumar, V.: Experimental testbed for large multirobot teams. Robot. Autom. Mag. IEEE **15**(1), 53–61 (2008)
5. Latronico, E., et al.: A vision of swarmlets. Internet Comput IEEE **19**(2), 20–28 (2015)

6. Leung, J.M.-K., et al.: Scalable semantic annotation using latticebased ontologies. In: 12th International Conference on Model Driven Engineering Languages and Systems (MODELS 09), pp. 393–407. ACM/IEEE (2009)
7. Lickly, B.: Static Model Analysis with Lattice-based Ontologies. Tech. rep. UCB/EECS-2012-212. Ph.D. Thesis. EECS Department, University of California, Berkeley (2012)
8. Lickly, B., et al.: A practical ontology framework for static model analysis. In: International Conference on Embedded Software (EMSOFT), ACM, pp. 23–32 (2011)
9. Fonseca, F., Davis, C., Câmara, G.: Bridging ontologies and conceptual schemas in geographic information integration. GeoInformatica **7**(4), 355–378 (2003)
10. Grenon, P., Smith, B.: SNAP and SPAN: towards dynamic spatial ontology. Spat. Cogn. Comput. **4**(1), 69–104 (2004)
11. Spaccapietra, S., et al.: On Spatial Ontologies (2004). https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.88.7653
12. Harris, S., Seaborne, A., Prud'hommeaux, E.: SPARQL 1.1 Query Language. (2013). https://www.w3.org/TR/sparql11-query/
13. Battle, R. Kolas, D.: Enabling the geospatial semantic web with parliament and GeoSPARQL". In: Semantic Web 3.4, pp. 355–370 (2012)
14. Lieberman, J., Singh, R., Goad, C.: W3C Geospatial Ontologies. Tech. rep. (2007)
15. Sun, K. et al.: Geospatial data ontology: the semantic foundation of geospatial data integration and sharing. In: Big Earth Data 3.3, pp. 269–296 (2019)
16. Butler, H., et al.: The GeoJSON Format. Tech. Rep. (2015). https://tools.ietf.org/html/draft-butler-geojson-06
17. Portele, C.: OpenGIS® Geography Markup Language (GML) Implementation Specification, version. In: (2007). https://portal.opengeospatial.org/files/?artifact_id=20509
18. Lee, J., et al.: OGC IndoorGML version 1.0. In: (2014). https://docs.opengeospatial.org/is/14-005r5/14-005r5.html
19. de Berg, M., et al.: Computational Geometry: Algorithms and Applications, 3rd edn. Springer TELOS, Santa Clara, CA, USA (2008)
20. Rusu, R.B., Cousins, S.: 3D is here: point cloud library (PCL). In: IEEE International Conference on Robotics and Automation (ICRA). Shanghai, China (2011)
21. Weiss, W., D'Mello, C.: Fundamentals of Model Theory. Topology Atlas (2000)
22. Kuipers, B.: The spatial semantic hierarchy. Artif. Intell. **119**(1-2), 191–233 (2000)
23. Ghrist, R., et al.: Topological landmark-based navigation and mapping. In: University of Pennsylvania, Department of Mathematics, Tech. Rep 8 (2012)
24. Moore, D. et al.: Robust distributed network localization with noisy range measurements. In: Proceedings of the 2nd international Conference on Embedded Networked Sensor Systems. ACM, pp. 50–61 (2004)
25. Weber, M. et al.: Gordian: formal reasoning based outlier detection for secure localization. Tech. rep. UCB/EECS-2019-1. EECS Department, University of California, Berkeley (2019)
26. Yang, Z., et al.: Beyond triangle inequality: sifting noisy and outlier distance measurements for localization. In: ACM Transactions on Sensor Networks 9.2, pp. 1–20 (2013)
27. Frank, Andrew U.: Qualitative spatial reasoning about distances and directions in geographic space. J. Vis. Lang. Comput. **3**(4), 343–371 (1992)
28. Cohn, A.G., Renz, J., et al.: Qualitative spatial representation and reasoning. In: Handbook of Knowledge Representation, vol. 3, pp. 551–596 (2008)
29. Pereira, E.T.: Mobile reactive systems over bigraphical machines a programming model and its implementation. Ph.D. Thesis, University of California, Berkeley (2015)
30. Atanasov, N., et al.: Semantic localization via the matrix permanent. Robot. Sci. Syst. (2014)
31. Vieu, L.: Spatial and temporal reasoning. In: Stock, O. (ed.) Spatial Representation and Reasoning in Artificial Intelligence, pp. 5–41. Springer, Netherlands, Dordrecht (1997)
32. Weber, M., Lee, E.: A model for semantic localization. In: ACM Press, pp. 350–351 (2015)