Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

b-it Bonn-Aachen International Center for Information Technology

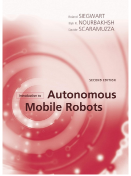
Institute for AI and Autonomous Systems

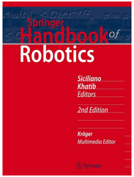
Environment Representations

How a Robot Models the World

Dr. Alex Mitrevski
Master of Autonomous Systems

Structure





- Importance of environment representations
- Environment representation types

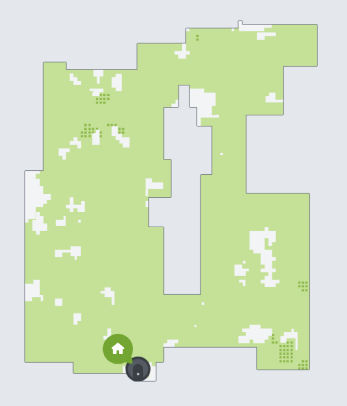
Importance of Environment Representations

What is an Environment Representation?

- An environment representation is a model $\mathcal{M}$ that **contains information about the environment in which a robot operates**
- Such a representation enables a robot to **maintain its belief about the state of the environment**
  - Reminder: We briefly introduced the belief in the lecture on perception
- Environment representations are often referred to as **world models** in the literature

An environment representation (aka a world model) encodes a robot's knowledge about its environment and enables the robot to model its belief about the environment's state

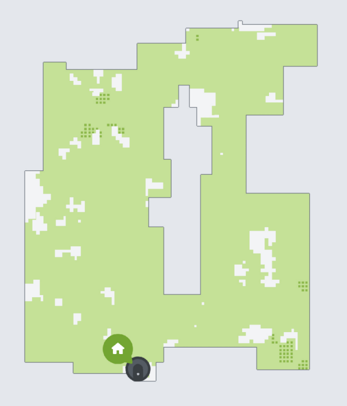
Why Does a Robot Need This?



https://homesupport.irobot.com/s/article/1467

- For an autonomous robot to move around in its environment, it needs to **have an understanding of environment in which it exists** — without that, it cannot effectively plan motions, avoid obstacles, and reach destinations

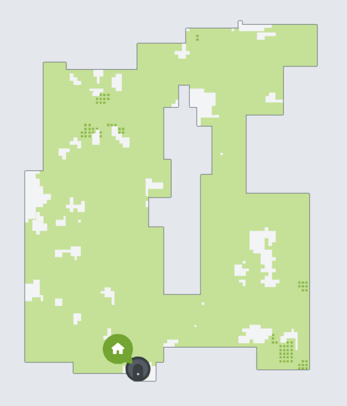
Why Does a Robot Need This?



https://homesupport.irobot.com/s/article/1467

- For an autonomous robot to move around in its environment, it needs to **have an understanding of environment in which it exists** — without that, it cannot effectively plan motions, avoid obstacles, and reach destinations
- Environment representations are not only useful for moving around though — **performing purposeful actions also requires knowledge of the environment**, such as objects that are present and actions that can be performed with them

Why Does a Robot Need This?



https://homesupport.irobot.com/s/article/1467

- For an autonomous robot to move around in its environment, it needs to **have an understanding of environment in which it exists** — without that, it cannot effectively plan motions, avoid obstacles, and reach destinations

- Environment representations are not only useful for moving around though — **performing purposeful actions also requires knowledge of the environment**, such as objects that are present and actions that can be performed with them

- **A robot without an appropriate environment representation has a limited applicability** because it does not have a memory about how the state of the world evolves
  - Vacuum cleaning robots are a good example of this: old models (or new, but cheap models) are either purely reactive or follow a set of preplanned motions; newer (more expensive models) can create cleaning maps for more effective operation

What Goes Into an Environment Representation?

- There are **multiple types of information that can be relevant** to model in an environment representation, such as:
  - **Obstacle locations and shapes**

What Goes Into an Environment Representation?

- There are **multiple types of information that can be relevant** to model in an environment representation, such as:
  - **Obstacle locations and shapes**
  - **Relevant landmarks** (e.g. unique patterns or objects on a wall)

What Goes Into an Environment Representation?

- There are **multiple types of information that can be relevant** to model in an environment representation, such as:
  - **Obstacle locations and shapes**
  - **Relevant landmarks** (e.g. unique patterns or objects on a wall)
  - **Locations or identities of objects**

What Goes Into an Environment Representation?

- There are **multiple types of information that can be relevant** to model in an environment representation, such as:
  - **Obstacle locations and shapes**
  - **Relevant landmarks** (e.g. unique patterns or objects on a wall)
  - **Locations or identities of objects**
- To represent such aspects, an environment representation can encode two types of information:

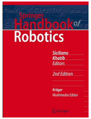
What Goes Into an Environment Representation?

- There are **multiple types of information that can be relevant** to model in an environment representation, such as:
  - **Obstacle locations and shapes**
  - **Relevant landmarks** (e.g. unique patterns or objects on a wall)
  - **Locations or identities of objects**
- To represent such aspects, an environment representation can encode two types of information:
  - **Geometric information**: This includes aspects such as **shapes**, **poses**, and **sizes**

What Goes Into an Environment Representation?

- There are **multiple types of information that can be relevant** to model in an environment representation, such as:
  - **Obstacle locations and shapes**
  - **Relevant landmarks** (e.g. unique patterns or objects on a wall)
  - **Locations or identities of objects**
- To represent such aspects, an environment representation can encode two types of information:
  - **Geometric information**: This includes aspects such as **shapes**, **poses**, and **sizes**
  - **Semantic information**: Examples of such information are **connections between objects or locations** (e.g. that two rooms are connected by a door), **object identities** (e.g. that a given object is a mug), and **semantic object locations** (e.g. a mug is on a kitchen counter)

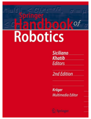
Requirements for Environment Representations



Compactness

A robot needs to consult and update its environment representation continuously, so **the representation should be encoded in a way that enables efficient processing**

Requirements for Environment Representations



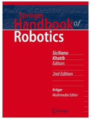
Compactness

A robot needs to consult and update its environment representation continuously, so **the representation should be encoded in a way that enables efficient processing**

Task specificity

The type of suitable representation differs between tasks, so **an appropriate representation (or a collection of representations) needs to be selected based on a robot's application domain**

Requirements for Environment Representations



Compactness

A robot needs to consult and update its environment representation continuously, so **the representation should be encoded in a way that enables efficient processing**

Task specificity

The type of suitable representation differs between tasks, so **an appropriate representation (or a collection of representations) needs to be selected based on a robot's application domain**

Uncertainty representation

Environment representations are created from sensor data and are updated based on sensor information extraction; both of these processes include noise and uncertainty, so **the representation should be designed to take this into account**

Global and Local Models

- The most general use of the term world model refers to **aggregated information about the complete environment**, but **environment representations can also represent more constrained environment aspects**

Global and Local Models

- The most general use of the term world model refers to **aggregated information about the complete environment**, but **environment representations can also represent more constrained environment aspects**
- In this respect, we can distinguish between **global and local models**:

Global and Local Models

- The most general use of the term world model refers to **aggregated information about the complete environment**, but **environment representations can also represent more constrained environment aspects**
- In this respect, we can distinguish between **global and local models**:
  - A **global environment model** encodes **information that represents the environment as a whole and which remains mostly static** (e.g. the locations of walls in a building)

Global and Local Models

- The most general use of the term world model refers to **aggregated information about the complete environment**, but **environment representations can also represent more constrained environment aspects**
- In this respect, we can distinguish between **global and local models**:
  - A **global environment model** encodes **information that represents the environment as a whole and which remains mostly static** (e.g. the locations of walls in a building)
  - A **local environment model** encodes **information about a small part of the environment and which changes more dynamically** (e.g. locations of people moving around a robot)

Global and Local Models

- The most general use of the term world model refers to **aggregated information about the complete environment**, but **environment representations can also represent more constrained environment aspects**
- In this respect, we can distinguish between **global and local models**:
  - A **global environment model** encodes **information that represents the environment as a whole and which remains mostly static** (e.g. the locations of walls in a building)
  - A **local environment model** encodes **information about a small part of the environment and which changes more dynamically** (e.g. locations of people moving around a robot)
- The distinction between global and local models is directly visible in the ROS navigation stack, where **a global costmap** is used for path planning (in a pre-created map), while a **local costmap** is used during instantaneous motion planning (in a local map that is continuously updated)

Representation Abstraction Levels

- Environment representations often represent information at a single level of abstraction; however, in many cases, **it can instead be more useful to use representations at multiple levels of abstraction**
  - For instance, general information about connectivity between rooms can be useful for room-to-room navigation, but more detailed representation of a room is necessary for navigation within a room

Representation Abstraction Levels

- Environment representations often represent information at a single level of abstraction; however, in many cases, **it can instead be more useful to use representations at multiple levels of abstraction**
  - For instance, general information about connectivity between rooms can be useful for room-to-room navigation, but more detailed representation of a room is necessary for navigation within a room
- Using **hierarchical representations** can enable a robot to switch between representations depending on the requirements of a concrete task

Representation Abstraction Levels

- Environment representations often represent information at a single level of abstraction; however, in many cases, **it can instead be more useful to use representations at multiple levels of abstraction**
  - For instance, general information about connectivity between rooms can be useful for room-to-room navigation, but more detailed representation of a room is necessary for navigation within a room
- Using **hierarchical representations** can enable a robot to switch between representations depending on the requirements of a concrete task
- Having multiple representations at different abstraction levels would, however, **increase the complexity of the model acquisition and update process**
  - The plasticity of different representations differs — some representations are more prone to being modified than others

Environment Representation Types

Representations Overview

- There is a variety of world model representations in robotics, the majority of which are **designed specifically for being used for navigation**

Representations Overview

- There is a variety of world model representations in robotics, the majority of which are **designed specifically for being used for navigation**
- In the navigation context, world models are typically referred to as **maps**
  - The process of creating or acquiring a map is called **mapping**

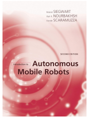
Representations Overview

- There is a variety of world model representations in robotics, the majority of which are **designed specifically for being used for navigation**
- In the navigation context, world models are typically referred to as **maps**
  - The process of creating or acquiring a map is called **mapping**
- Specifically for maps, we can distinguish between two major types — **metric** and **topological**

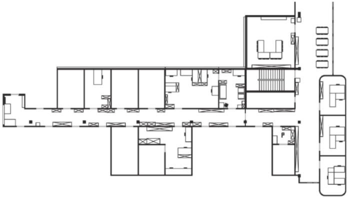
Representations Overview

- There is a variety of world model representations in robotics, the majority of which are **designed specifically for being used for navigation**
- In the navigation context, world models are typically referred to as **maps**
  - The process of creating or acquiring a map is called **mapping**
- Specifically for maps, we can distinguish between two major types — **metric** and **topological**
- Beyond navigation, **semantic world models** can also be useful

Metric Maps

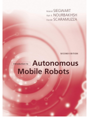


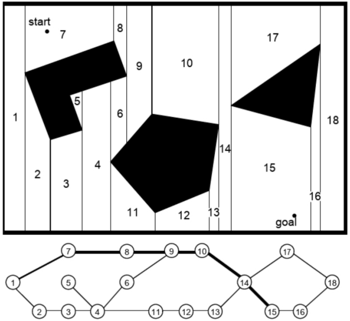
- Metric maps **encode geometric information about an environment**
- The representation can be in a **continuous form** (e.g. linear features or polygons) or a **discrete form** (e.g. grid cells)
- Metric maps **abstract away non-geometric environmental features** — for instance, information about the color or texture of obstacles is not included in a metric map
- The information density of metric maps typically makes them **suitable for being used for obstacle avoidance** — obstacles are geometric entities that need to be avoided during navigation



Exact Cell Decomposition

Metric Maps

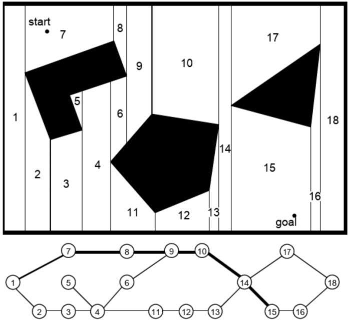




- An **exact cell decomposition** is a type of metric map that is created by **splitting space based on object vertices**
  - Objects thus need to be represented by polygons

Exact Cell Decomposition

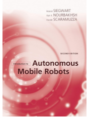
Metric Maps

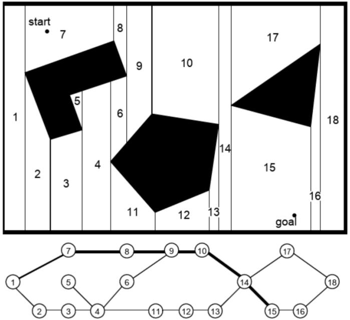


- An **exact cell decomposition** is a type of metric map that is created by **splitting space based on object vertices**
  - Objects thus need to be represented by polygons
- The split of cells is typically done by:
  1. **sorting obstacles along the x-axis** and
  2. **creating a cell boundary passing through each vertex** (from left to right)

Exact Cell Decomposition

Metric Maps

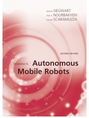


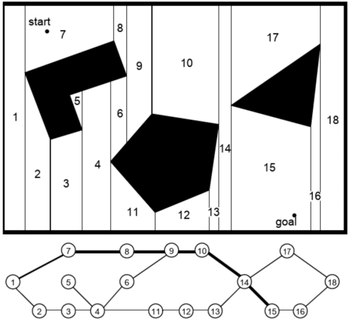


- An **exact cell decomposition** is a type of metric map that is created by **splitting space based on object vertices**
  - Objects thus need to be represented by polygons
- The split of cells is typically done by:
  1. **sorting obstacles along the x-axis** and
  2. **creating a cell boundary passing through each vertex** (from left to right)
- This results in a **full decomposition of the environment** that is suitable to use for path planning

Exact Cell Decomposition

Metric Maps

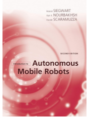




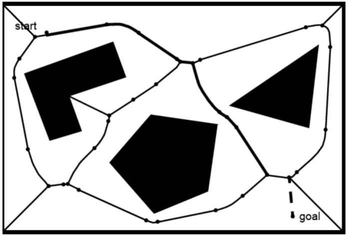
- An **exact cell decomposition** is a type of metric map that is created by **splitting space based on object vertices**
  - Objects thus need to be represented by polygons
- The split of cells is typically done by:
  1. **sorting obstacles along the x-axis** and
  2. **creating a cell boundary passing through each vertex** (from left to right)
- This results in a **full decomposition of the environment** that is suitable to use for path planning
- An exact cell decomposition requires knowledge of the geometries of all objects — typically something that is not known

Voronoi Diagram

Metric Maps

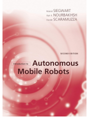


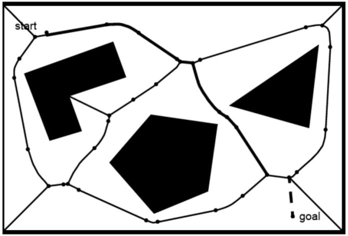
- A **Voronoi diagram** is conceptually similar to the exact cell decomposition in terms of its outcomes, but differs in the sense that **it splits space by maximising distances to objects**



Voronoi Diagram

Metric Maps

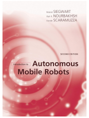


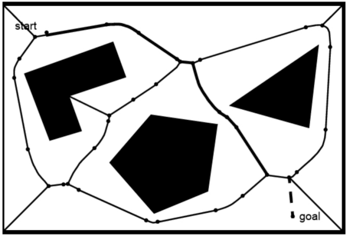


- A **Voronoi diagram** is conceptually similar to the exact cell decomposition in terms of its outcomes, but differs in the sense that **it splits space by maximising distances to objects**
- By definition, a Voronoi decomposition considers n points $\boldsymbol{x}_i, 1 \leq i \leq n$ and creates n regions $V_i, 1 \leq i \leq n$, such that V_i **contains all points that are closest to** x_i

Voronoi Diagram

Metric Maps

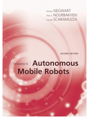




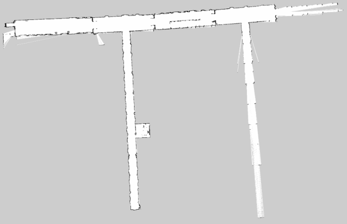
- A **Voronoi diagram** is conceptually similar to the exact cell decomposition in terms of its outcomes, but differs in the sense that **it splits space by maximising distances to objects**
- By definition, a Voronoi decomposition considers n points $\boldsymbol{x}_i, 1 \leq i \leq n$ and creates n regions $V_i, 1 \leq i \leq n$, such that V_i **contains all points that are closest to** $\boldsymbol{x}_i$
- For a Voronoi decomposition around objects, **edges are created that are equidistant from two or more obstacles**; such edges contain both straight-line and curved segments:
  - straight-line segments are defined between **points that are equidistant from two lines** and
  - curved (parabolic) edges are defined for **points that are equidistant from a line and an object corner**

2D Occupancy Grid

Metric Maps

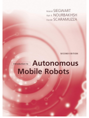


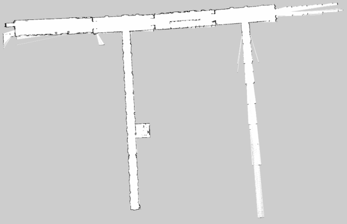
- A 2D occupancy grid is a two-dimensional discrete map that **represents an environment by a set of grid cells, each of which has an occupancy probability**



2D Occupancy Grid

Metric Maps

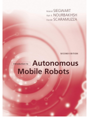


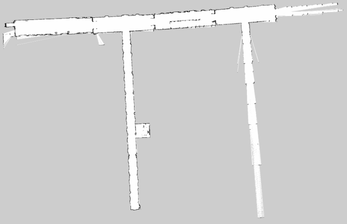


- A 2D occupancy grid is a two-dimensional discrete map that **represents an environment by a set of grid cells, each of which has an occupancy probability**
- We can define an occupancy grid to have the form $\mathcal{M}^{OC} = (w, h, r, o_x, o_y, OC)$, where:
  - r is the **grid cell resolution** (e.g. in meters)
  - w and h are the **map's width and height**, respectively
  - o_x and o_y represent the **coordinates of the map's origin point** (in continuous coordinates)
  - $OC \in \mathbb{R}^{h \times w}$ **is an occupancy probability matrix**, namely $OC_{i,j} \in [0, 1], 1 \leq i \leq h, 1 \leq j \leq w$

2D Occupancy Grid

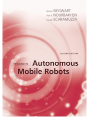
Metric Maps

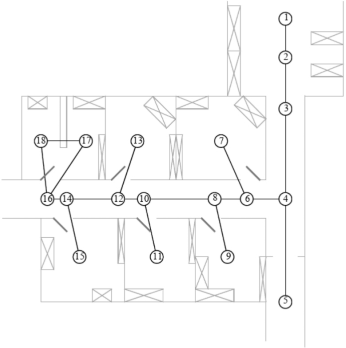




- A 2D occupancy grid is a two-dimensional discrete map that **represents an environment by a set of grid cells, each of which has an occupancy probability**
- We can define an occupancy grid to have the form $\mathcal{M}^{OC} = (w, h, r, o_x, o_y, OC)$, where:
  - r is the **grid cell resolution** (e.g. in meters)
  - w and h are the **map's width and height**, respectively
  - o_x and o_y represent the **coordinates of the map's origin point** (in continuous coordinates)
  - $OC \in \mathbb{R}^{h \times w}$ **is an occupancy probability matrix**, namely $OC_{i,j} \in [0, 1], 1 \leq i \leq h, 1 \leq j \leq w$
- 2D occupancy grid can be acquired from 2D measurements and are very commonly used in (indoor) robotics
  - We will discuss 2D mapping later in the course

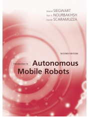
Topological Maps

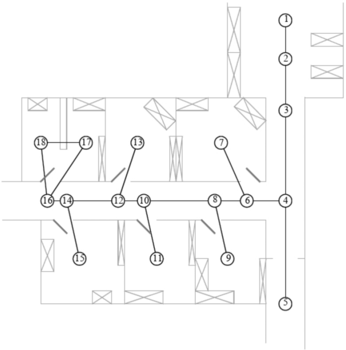




- Unlike metric maps, **topological maps encode information about connectivity between locations**

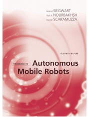
Topological Maps

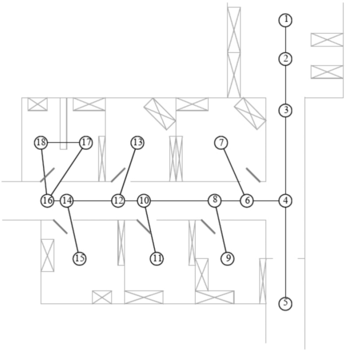




- Unlike metric maps, **topological maps encode information about connectivity between locations**
- A topological map is **represented as a weighted graph** $\mathcal{M}^T = (N, V, w)$ in which:
  - the nodes N represent **locations**
  - the edges $V : N \rightarrow N$ are **connections between locations**
  - $w : V \rightarrow \mathbb{R}$ are **weights associated with the edges**

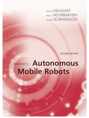
Topological Maps

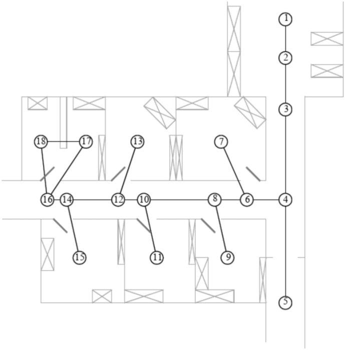




- Unlike metric maps, **topological maps encode information about connectivity between locations**
- A topological map is **represented as a weighted graph** $\mathcal{M}^T = (N, V, w)$ in which:
  - the nodes N represent **locations**
  - the edges $V : N \rightarrow N$ are **connections between locations**
  - $w : V \rightarrow \mathbb{R}$ are **weights associated with the edges**
- Topological maps do not encode any metric information about the environment — **locations are modelled as points**

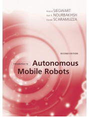
Topological Maps





- Unlike metric maps, **topological maps encode information about connectivity between locations**
- A topological map is **represented as a weighted graph** $\mathcal{M}^T = (N, V, w)$ in which:
  - the nodes N represent **locations**
  - the edges $V : N \rightarrow N$ are **connections between locations**
  - $w : V \rightarrow \mathbb{R}$ are **weights associated with the edges**
- Topological maps do not encode any metric information about the environment — **locations are modelled as points**
- Due to their graph structure, **topological maps are suitable for efficient path planning**

Relation Between Map and State Representations



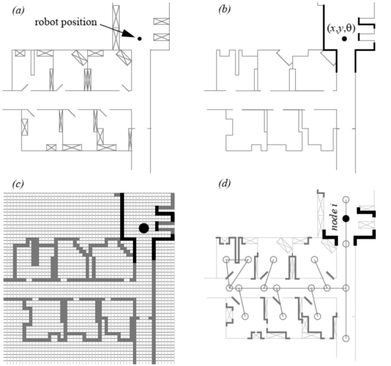
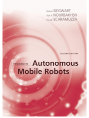


Figure 5.10
Three examples of single hypotheses of position using different map representations: (a) real map with walls, doors and furniture; (b) line-based map → around 100 lines with two parameters; (c) occupancy grid-based map → around 3000 grid cells size 50 × 50 cm; (d) topological map using line features (Z/S lines) and doors → around 50 features and 18 nodes.

- The information encoded by different types of maps differs in nature — and so do **the statements that can be made about the robot's state within such a map**
  - A metric map can be used to **define a robot's pose in continuous coordinates**
  - A topological map determines **the position of a robot in terms of a graph node**

Relation Between Map and State Representations



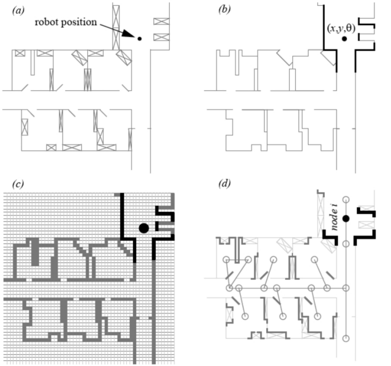
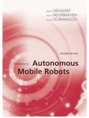


Figure 5.10
Three examples of single hypotheses of position using different map representations: (a) real map with walls, doors and furniture; (b) line-based map → around 100 lines with two parameters; (c) occupancy grid-based map → around 3000 grid cells size 50 × 50 cm; (d) topological map using line features (Z/S lines) and doors → around 50 features and 18 nodes.

- The information encoded by different types of maps differs in nature — and so do **the statements that can be made about the robot's state within such a map**
  - A metric map can be used to **define a robot's pose in continuous coordinates**
  - A topological map determines **the position of a robot in terms of a graph node**
- The different representations also influence **the environmental aspects that are modelled**
  - Metric maps can **encode precise information about obstacles in the environment**
  - Topological maps can more easily **encode information about paths that should be allowed or blocked**

Relation Between Map and State Representations



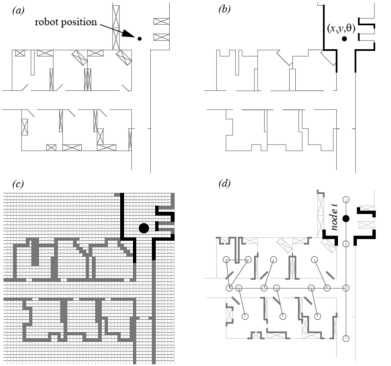
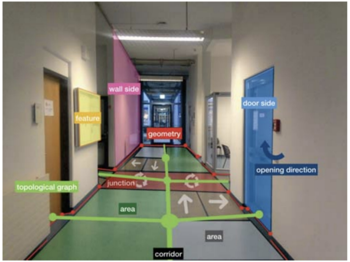


Figure 5.10
Three examples of single hypotheses of position using different map representations: (a) real map with walls, doors and furniture; (b) line-based map → around 100 lines with two parameters; (c) occupancy grid-based map → around 3000 grid cells size 50 × 50 cm; (d) topological map using line features (Z/S lines) and doors → around 50 features and 18 nodes.

- The information encoded by different types of maps differs in nature — and so do **the statements that can be made about the robot's state within such a map**
  - A metric map can be used to **define a robot's pose in continuous coordinates**
  - A topological map determines **the position of a robot in terms of a graph node**
- The different representations also influence **the environmental aspects that are modelled**
  - Metric maps can **encode precise information about obstacles in the environment**
  - Topological maps can more easily **encode information about paths that should be allowed or blocked**
- In practical applications, it can be useful to use metric and topological representations together

Semantic Map

Hybrid Maps

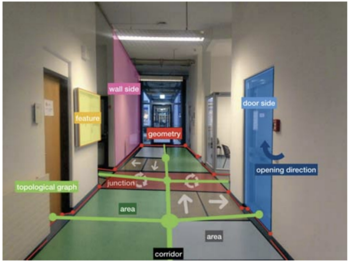


L. Naik et al., "Semantic mapping extension for OpenStreetMap applied to indoor robot navigation," in *Proc. Int. Conf. Robotics and Automation (ICRA)*, 2019, pp. 3839–3845.

- One way to combine metric and topological maps is to **combine them into a hierarchical map representation**

Semantic Map

Hybrid Maps

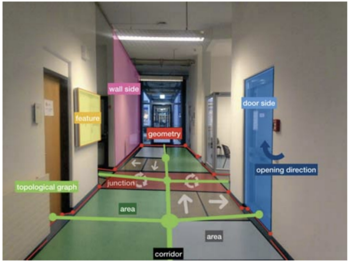


L. Naik et al., "Semantic mapping extension for OpenStreetMap applied to indoor robot navigation," in *Proc. Int. Conf. Robotics and Automation (ICRA)*, 2019, pp. 3839–3845.

- One way to combine metric and topological maps is to **combine them into a hierarchical map representation**
- A **semantic map** is one such type of map, which encodes:
  - connections between locations
  - geometric information about locations (e.g. room size) and
  - additional features that may be relevant during navigation (e.g. which side to move on)

Semantic Map

Hybrid Maps

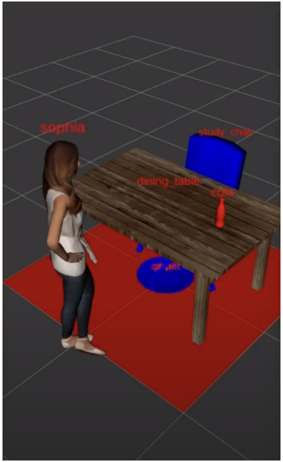


L. Naik et al., "Semantic mapping extension for OpenStreetMap applied to indoor robot navigation," in *Proc. Int. Conf. Robotics and Automation (ICRA)*, 2019, pp. 3839–3845.

- One way to combine metric and topological maps is to **combine them into a hierarchical map representation**
- A **semantic map** is one such type of map, which encodes:
  - connections between locations
  - geometric information about locations (e.g. room size) and
  - additional features that may be relevant during navigation (e.g. which side to move on)
- Semantic maps of this type can be **challenging to create** (they typically require considerable manual effort), but include **rich information that can be beneficial for both path and motion planning**

Symbolic World Representations

Beyond Navigation

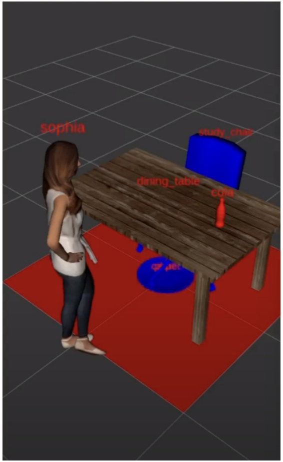


https://github.com/b-it-bots/rviz_3d_object_visualizer

- World models are not useful only for navigation — **(mobile) manipulation and human-robot collaboration also require appropriate environment representations**

Symbolic World Representations

Beyond Navigation

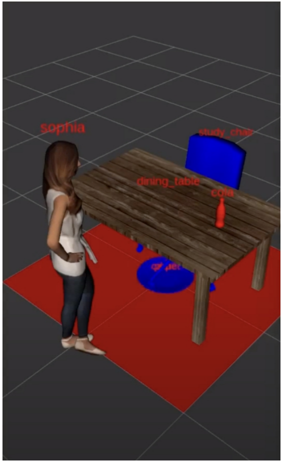


https://github.com/b-it-bots/rviz_3d_object_visualizer

- World models are not useful only for navigation — **(mobile) manipulation and human-robot collaboration also require appropriate environment representations**
- A **symbolic world model** is one commonly used representation that **encodes information about the environment through logical predicates**, e.g. `on(bottle, dining_table)`

Symbolic World Representations

Beyond Navigation

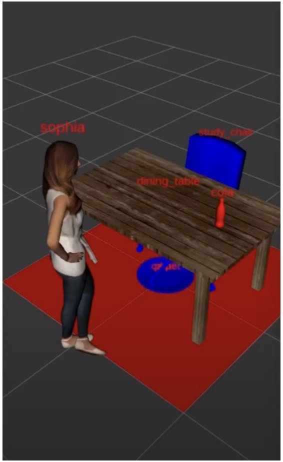


https://github.com/b-it-bots/rviz_3d_object_visualizer

- World models are not useful only for navigation — **(mobile) manipulation and human-robot collaboration also require appropriate environment representations**
- A **symbolic world model** is one commonly used representation that **encodes information about the environment through logical predicates**, e.g. on(bottle, dining_table)
- Such a representation **abstracts away geometric information** and can be used for **creating high-level task plans**

Symbolic World Representations

Beyond Navigation



https://github.com/b-it-bots/rviz_3d_object_visualizer

- World models are not useful only for navigation — **(mobile) manipulation and human-robot collaboration also require appropriate environment representations**
- A **symbolic world model** is one commonly used representation that **encodes information about the environment through logical predicates**, e.g. $\mathrm{on}(\texttt{bottle}, \texttt{dining\_table})$
- Such a representation **abstracts away geometric information** and can be used for **creating high-level task plans**
- Symbolic world models are typically **combined with geometric world models**; such a hybrid representation enables task plan execution

Neural Representations

- The "classical" environment representations discussed thus far require **a fair amount of modelling effort** (e.g. occupancy grids may require experimenting with the resolution to obtain reasonable precision without sacrificing efficiency)

Neural Representations

- The "classical" environment representations discussed thus far require **a fair amount of modelling effort** (e.g. occupancy grids may require experimenting with the resolution to obtain reasonable precision without sacrificing efficiency)
- Learning-based representations, particularly those based on neural networks, enable **acquiring environment representations from data**
  - In this case, **the environment representation is contained within the learned network connections**
  - Navigation using such a learned representation is referred to as **mapless navigation**

Neural Representations

- The "classical" environment representations discussed thus far require **a fair amount of modelling effort** (e.g. occupancy grids may require experimenting with the resolution to obtain reasonable precision without sacrificing efficiency)
- Learning-based representations, particularly those based on neural networks, enable **acquiring environment representations from data**
  - In this case, **the environment representation is contained within the learned network connections**
  - Navigation using such a learned representation is referred to as **mapless navigation**
- There are some key differences between classical and learned representations that need to be mentioned though:
  - Classical representations are **easily interpretable and lead to provably correct algorithms** (but are sometimes inefficient and based on simplifying assumptions)

Neural Representations

- The "classical" environment representations discussed thus far require **a fair amount of modelling effort** (e.g. occupancy grids may require experimenting with the resolution to obtain reasonable precision without sacrificing efficiency)

- Learning-based representations, particularly those based on neural networks, enable **acquiring environment representations from data**
  - In this case, **the environment representation is contained within the learned network connections**
  - Navigation using such a learned representation is referred to as **mapless navigation**

- There are some key differences between classical and learned representations that need to be mentioned though:
  - Classical representations are **easily interpretable and lead to provably correct algorithms** (but are sometimes inefficient and based on simplifying assumptions)
  - Learned world representations, on the other hand, are implicit, so it is typically **difficult to understand what they encode and whether they are guaranteed to lead to correct behaviour**

Neural Representations

- The "classical" environment representations discussed thus far require **a fair amount of modelling effort** (e.g. occupancy grids may require experimenting with the resolution to obtain reasonable precision without sacrificing efficiency)
- Learning-based representations, particularly those based on neural networks, enable **acquiring environment representations from data**
  - In this case, **the environment representation is contained within the learned network connections**
  - Navigation using such a learned representation is referred to as **mapless navigation**
- There are some key differences between classical and learned representations that need to be mentioned though:
  - Classical representations are **easily interpretable and lead to provably correct algorithms** (but are sometimes inefficient and based on simplifying assumptions)
  - Learned world representations, on the other hand, are implicit, so it is typically **difficult to understand what they encode and whether they are guaranteed to lead to correct behaviour**
- So far, classical environment representations remain dominant, particularly in the context of navigation