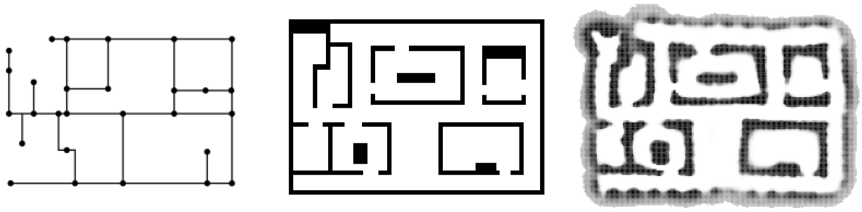
Roadmaps

⇛ A planner plans a path from a particular start configuration to a particular goal configuration.

⇛ If we knew that many paths were to be planned in the same environment, then it would make sense to construct a data structure once and then use that data structure to plan subsequent paths more quickly.

⇛ This data structure is often called a map, and mapping is the task of generating models of robot environments from sensor data.

⇛ In the context of indoor systems, three map concepts prevail:



1. Topological 2. Geometric 3. Grids

Topological

↳ Aim at representing environments with graphlike structures, where nodes correspond to "something distinct" and edges represent an adjacency relationship between nodes.

Geometric

↳ Geometric models use geometric primitives for representing the environment.

Grids

↳ Grid structures where value of each pixel corresponds to the likelihood that its corresponding portion of workspace or configuration space is occupied

↳ Occupancy grids were first introduced for mapping unknown spaces with wide-angle ultrasonic sensors

⇛ This chapter focuses on a class of topological maps called roadmaps.

⇛ A roadmap is embedded in the free space and hence the nodes and edges of a roadmap also carry physical meaning.

⇛ Robots use roadmaps in much the same way people use highway systems.

⇛ Instead of planning every possible side-street path to a destination, people usually plan their path to a network of highways, then along the highway system, and finally from the highway to their destination.

⇛ The bulk of the motion occurs on the highway system.

⇛ Likewise, using a roadmap, the planner can construct a path between any two points in a connected component of the robot's free space by:

1. First finding a collision-free path onto the roadmap.
2. Traversing the roadmap to the vicinity of the goal.
3. Then constructing a collision-free path from a point on the roadmap to the goal.

⇛ The bulk of the motion occurs on the roadmap and thus searching does not occur in a multidimensional space.

⇛ If the robot knows the roadmap, then it in essence knows the environment.

DEFINITION 5.0.2 (Roadmap) *A union of one-dimensional curves is a* ***roadmap*** RM *if for all* q_{start} *and* q_{goal} *in* $\mathcal{Q}_{\text{free}}$ *that can be connected by a path, the following properties hold:*

1. ***Accessibility***: *there exists a path from* $q_{\text{start}} \in \mathcal{Q}_{\text{free}}$ *to some* $q'_{\text{start}} \in RM$,
2. ***Departability***: *there exists a path from some* $q'_{\text{goal}} \in RM$ *to* $q_{\text{goal}} \in \mathcal{Q}_{\text{free}}$, *and*
3. ***Connectivity***: *there exists a path in* RM *between* q'_{start} *and* q'_{goal}.

⇛ In this chapter, we consider five types of roadmaps:

1. Visibility maps
2. Deformation retracts
3. Retract-like structures
4. Piecewise retracts
5. Silhouettes

1. Visibility maps

⇛ The defining characteristics of a visibility map are that:

→ Its nodes share an edge if they are within line of sight of each other.

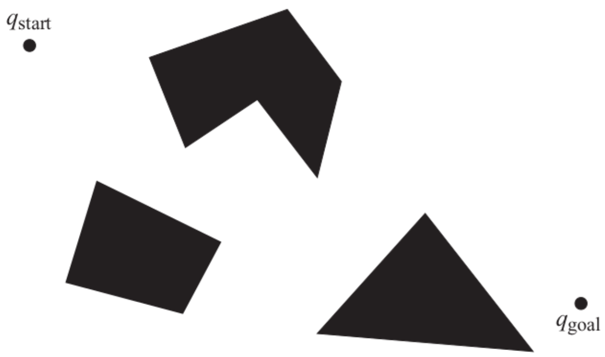
→ All points in the robot's free space are within line of sight of at least one node on the visibility map.

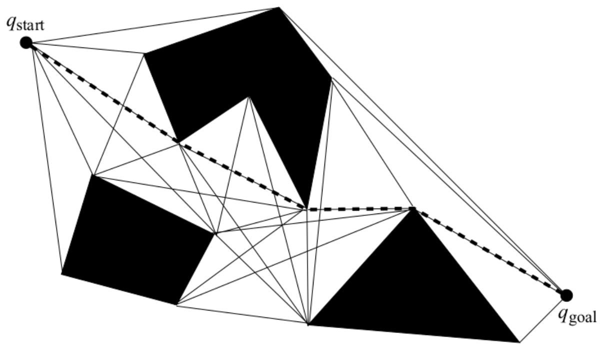
⇛ In this section, we consider the simplest visibility map, called the visibility graph.

• Visibility Graph Defination

⇛ The standard visibility graph is defined in a two-dimensional polygonal configuration space.

⇛ The nodes v_i of the visibility graph include the start location, the goal location, and all the vertices of the configuration space obstacles.





The graph edges e_{ij} are straight-line segments that connect two line-of-sight nodes v_i and v_j.

$$\boxed{e_{ij} \neq \emptyset \iff s v_i + (1-s) v_j \in \mathrm{cl}(\mathcal{Q}_{\text{free}}) \ \ \forall s \in [0, 1]}$$

$$\Updownarrow$$

$$\boxed{e_{ij} \neq \phi \iff v_i + s(v_j - v_i) \in cl(Q_{free}) \ \ \forall s \in [0, 1]}$$

Generally Q_free does not include contact betweent robot and obstacle.

cl(Q_free) denotes closure of Q_free i.e. it includes contact between robot and obstacle.

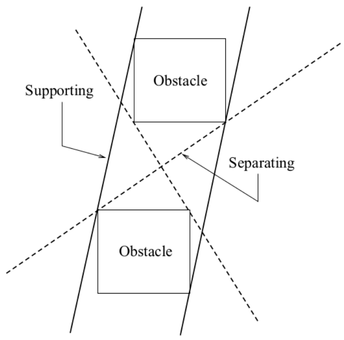
- Using the standard two-norm (Euclidean distance), the visibility graph can be searched for the shortest path

- The visibility graph can be defined for a three dimensional configuration space populated with polyhedral obstacles.
  - But it does not necessarily contain the shortest paths in such a space.

- Unfortunately, the visibility graph has many needless edges.
  - The use of supporting and separating lines can reduce the number of edges.

- A supporting line is tangent to two obstacles such that both obstacles lie on the same side of the line.

- A separating line is tangent to two obstacles such that the obstacles lie on opposite sides of the line.



- The reduced visibility graph is soley constructed from supporting and separating lines.

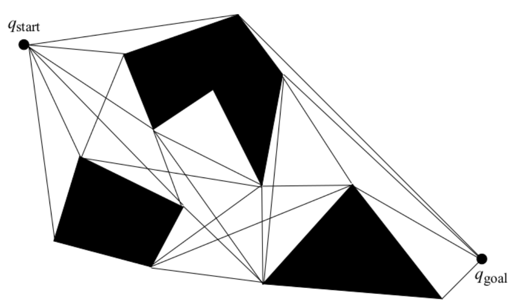
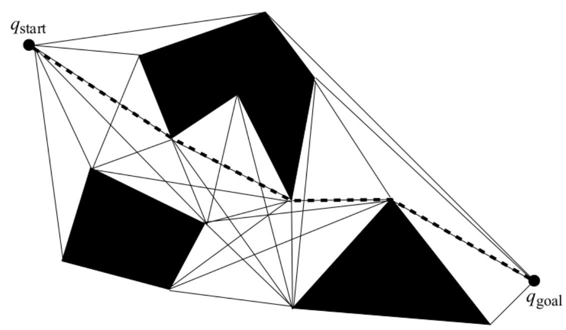


Figure 5.6 Reduced visibility graph.



Original graph

- The definitions of the supporting and separating lines applies to nonconvex shapes as well.

Visibility Graph Construction

- Let V = {v_1 , . . . , v_n } be the set of vertices of the polygons in the configuration space as well as the start and goal configurations.

- To construct the visibility graph, for each v ∈ V we must determine which other vertices are visible to v.

- The most obvious way to make this determination is to test all line segments $\overline{vv_i}$, $v \neq v_i$ to see if they intersect an edge of any polygon.

- For a particular $\overline{vv_i}$, there are O(n) intersections to check because there are O(n) edges from the obstacles.

- Now, there are O(n) potential segments emanating from v, so for a particular v, there are O(n^2) tests to determine which vertices are indeed visible from v.

- This must be done for all v ∈ V and thus the construction of the visibility graph would have complexity O(n^3).

- There is a more efficient way to compute the set of vertices that are visible from v, known as plane sweep algorithms.

- A plane sweep algorithm solves a problem by sweeping a line, called the sweep line, across the plane, pausing at each of the vertices of the obstacles.

- At each vertex, the algorithm updates a partial solution to the problem.

- For the problem of computing the set of vertices visible from v, we will let the sweep line, l, be a half-line emanating from v, and we will use a rotational sweep, rotating l from 0 to 2π .

Algorithm 5 Rotational Plane Sweep Algorithm

Input: A set of vertices $\{v_i\}$ (whose edges do not intersect) and a vertex v

Output: A subset of vertices from $\{v_i\}$ that are within line of sight of v

```
For each vertex v_i, calculate α_i, the angle from the horizontal axis to the line segment vv_i.
Create the vertex list 𝓔, containing the α_i's sorted in increasing order.
Create the active list 𝓢, containing the sorted list of edges that intersect the horizontal half-line emanating from v.
for all α_i do
    if v_i is visible to v then
        Add the edge (v, v_i) to the visibility graph.
    end if
    if v_i is the beginning of an edge, E, not in 𝓢 then
        Insert the E into 𝓢.
    end if
    if v_i is the end of an edge in 𝓢 then
        Delete the edge from 𝓢.
    end if
end for
```

By checking if it intersect the edges in active list S.

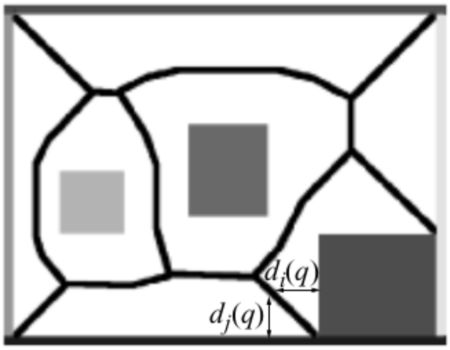
- The set S is incrementally constructed as the algorithm runs.

- The complexity of the algorithm is O(n^2 log n).

- Finally, we have not considered here the case when l may simultaneously intersect multiple vertices.
  - When this does occur, the problem can be resolved by slightly perturbing the position of one of the three vertices.

2. Deformation Retracts: Generalized Voronoi Diagram (GVD)

⇒ The generalized Voronoi diagram (GVD) is the set of points where the distance to the two closest obstacles is the same.



⇒ Path planning is achieved by moving away from the closest point until reaching the GVD, then along the double equidistant GVD to the vicinity of the goal, and then from the GVD to the goal.

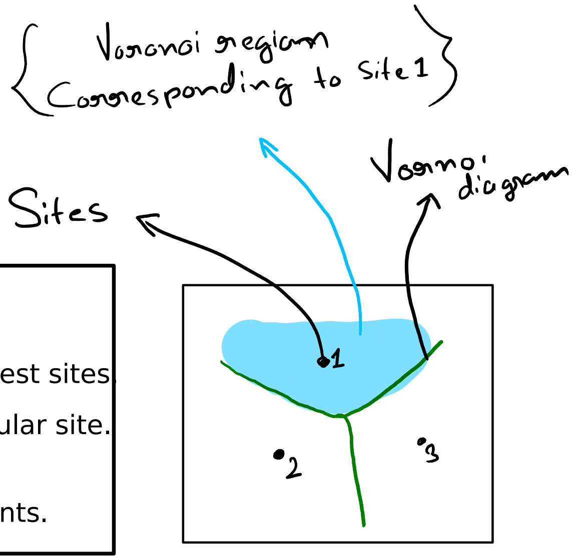
⇒ Since the GVD is defined in terms of distance, one can expect that a robot equipped with range sensors can incrementally construct the GVD in an unknown space.

⇒ Once the GVD is constructed, the robot has essentially explored the space because the robot can use the GVD to plan paths in the free space with the GVD.

⇒ GVD is a type of deformation retract.

• GVD Definition

⇒ The Voronoi diagram is defined for a set of points called sites.
⇒ A Voronoi region is the set of points closest to a particular site.
⇒ The Voronoi diagram is then the set of points equidistant to two closest sites.
⇒ It sections off the free space into regions that are closest to a particular site.
⇒ Points on the Voronoi diagram have two closest sites.
⇒ In the planar case, the Voronoi diagram is a collection of line segments.



⇒ For the purposes of path planning, we can think of the point sites as obstacles, but obstacles are not simple points.

⇒ Therefore, the definition of a Voronoi region is extended to the generalized Voronoi region, F_i , which is the closure of the set of points closest to QO_i .

$$\mathcal{F}_i = \{q \in \mathcal{Q}_{\text{free}} \mid d_i(q) \leq d_h(q) \quad \forall h \neq i\}$$

↳ where $d_i(q)$ is the distance to an obstacle $\mathcal{QO}_i$ from q, i.e., $d_i(q) = \min_{c \in \mathcal{QO}_i} d(q, c)$

⇒ The basic building block of the GVD is the set of points equidistant to two sets $\mathcal{QO}_i$ and $\mathcal{QO}_j$, which we term a *two-equidistant surface* denoted by $\mathcal{S}_{ij} = \{x \in \mathcal{Q} \mid (d_i(q) - d_j(q)) = 0\}$.

⇛ A two-equidistant surface pierces obstacles, so we restrict it to the set of points that are both equidistant to QO_i and QO_j and have QO_i and QO_j as their closest obstacles.

⇛ This restricted structure is the two-equidistant face, which could be denoted by

$$\mathcal{F}_{ij} = \{q \in \mathcal{S}_{ij} \mid d_i(q) \leq d_h(q) \ \ \forall h\}^2.$$

⇛ The union of the two-equidistant faces forms the GVD.

$$\mathrm{GVD} = \bigcup_i \bigcup_j \mathcal{F}_{ij}$$

⇛ This definition of the GVD applies to any dimensional spaces.

⇛ Meet points: The set of points equidistant to three or more obstacles.

⇛ Boundary points: The set of points whose distance to the closest obstacle is zero.

• GVD Roadmap Properties

⇛ In $\mathbb{R}^m$, the GVD has the properties of accessibility, connectivity, and departability.

⇛ In the plane, the GVD is a roadmap because it has these properties and is one-dimensional.

⇛ The robot achieves accessibility by moving away from the closest obstacle.

↳ It performs gradient ascent of distance D to the closest obstacle

$$\frac{dc(t)}{dt} = \nabla D(c(t)) \quad \text{where} \ \ c(0) = q_{\text{start}}.$$

until it reaches a point on the GVD.

⇛ In an obstacle-bounded environment, gradient ascent of D traces a path from any point in the free space to the GVD.

⇛ A deformation retract is the image of a continuous function called a deformation retraction RM such that

$$RM(q) = q, \quad \text{for all } q \text{ in the GVD},$$
$$RM(q) = q', \quad \text{for any } q \in \mathcal{Q}_{\text{free}} \text{ and } q' \in \text{GVD}$$

$$\boxed{q'_{\text{start}} = RM(q_{\text{start}}) \text{ and } q'_{\text{goal}} = RM(q_{\text{goal}})}$$

⇛ Departability is simply accessibility in reverse.

Deformation Retract Definition

⇒ Before defining the deformation retract, we define a weaker structure called a retract.

⇒ For a manifold X, a retraction is a continuous function $f : X \to A$ such that $A \subset X$, and $f(a) = a$ for all $a \in A$.

↳ The subset A is the retract.

⇒ The set of deformation retracts is a subset of the set of retracts and hence the GVD is a retract also.

⇒ A deformation retract inherits many topological properties from its ambient space, whereas a retract may not.

↳ One important property is that the number of "types" of closed paths in the free space is equal to the number of "types" of closed paths in the deformation retract of the free space.

⇒ Let $f : U \to V$ and $g : U \to V$ where U and V are manifolds.

⇒ A homotopy is a continuous function $H : U \times [0, 1] \to V$ such that $H(x, 0) = f(x)$ and $H(x, 1) = g(x)$.

→ An example of H is $H(x, t) = (1 - t) f(x) + t g(x)$.

→ If there exists such a continuous mapping that "deforms" f to g, then f and g are homotopic.

→ resulting equivalence relation is denoted $f \sim g$.

⇒ We can also say that two paths f and g are path-homotopic, i.e., f ~ g, if they can be continuously deformed into one another.

⇒ This relation allows for the classification of functions into equivalence classes termed path-homotopy classes and are denoted as

$$[c] = \{\bar{c} \in C^0 \mid \bar{c} \sim c\}$$

⇒ Let $A \subset X$ and let $f : X \rightarrow A$ be a retraction. A *deformation retraction* is a homotopy $H : X \times [0, 1] \rightarrow X$ such that

- $H(x, 0) = x$
- $H(x, 1) \in A$
- $H(a, t) = a$ for $a \in A$ and $t \in [0, 1]$

⇒ In other words, H is a homotopy between a retraction and the identity map.

→ Note that all retractions are not necessarily homotopic to the identity map.

→ The retract is now called a deformation retract.

⇒ We use deformation retractions to smoothly deform, without tearing or pasting X onto a lower, preferably one-dimensional subset A of X .

→ So, as t varies from 0 to 1, a point in X continuously moves through X to a point in A.

→ Thefore, the deformation retraction preserves many topological properties of the free space.

⇒ While a diffeomorphism preserves the structure of two spaces of the same dimension, a deformation retraction preserves the structure of two spaces of different dimension.

⇒ One of the key topological properties of deformation retracts is that they preserve the number of homotopically equivalent closed loops from the ambient space.

⇒ The number of homotopy equivalence classes of closed loops is called the first fundamental group, and is denoted as π_1(X, x_0) for loops in X passing through x_0 .

→ Since this is a group, it has a group operator ★) that simply concatenates paths.

⇒ If f is a deformation retraction with A as its deformation retract of X , then

$$\pi_1(X, x_0) = \pi_1(A, f(x_0))$$

→ In other words, the ambient space X and the deformation retract A have the same number of homotopically equivalent closed loops.

⇒ Deformation retracts have the properties of connectivity, accessibility, and departability.

⇒ Let H be the deformation retraction and H (x, 0) = q_start .

⇒ The path to the deformation retract is then defined by H (x, ·) : [0, 1] → Q free where H (x, 1) is an element of the deformation retract.

→ Departability is shown in the same manner.

⇒ Since the deformation retract is connected, there is a path between the retracted start and retracted goal configurations along the deformation retract.

⇒ Hence, one-dimensional deformation retracts are roadmaps.

• GVD Dimension {The Preimage Theorem and Critical Points}

A key property of a roadmap is that it is one-dimensional.

We are using the distance function d i to define the GVD, but this function assumes that the obstacles are convex, which is unrealistic in most situations.

At first, it seems to make sense to decompose nonconvex obstacles into convex pieces.

This causes problems because there are many ways to construct such a decomposition, thereby resulting in different representations of the free space.

It would be nice to have a unique representation of the roadmap, so we refine our definition of the GVD.