5

Roadmaps (Cont.)

★ Deformation Retract Defination

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

retraction

↳ It is a continuous function $f: X \to A$ such that $A \subset X$ & $f(a) = a \ \forall a \in A$.

$\{X$ is a manifold$\}$

↳ The subset A is the retract.

↳ Typically, the dimension of A is less than the dimension of X.

⇒ GVD is also a retract.

⇒ However, the properties of a retract are not sufficient to guarantee that GVD is a roadmap.

↳ It is the fact that GVD is indeed a deformation retract that makes it a roadmap.

⇒ 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 ~~the~~ free space is equal to the number of types of closed paths in the deformation retract of the free space.

⇒ We need to enforce additional properties on the retract so ~~as~~ to guarantee that it captures the topology of its free space and ~~still~~ still ~~and~~ one-dimensional.

Global diffeomorphisms

↳ Mappings that relate spaces that are "topologically similar"

⇒ Let us consider spaces that are similar but of different dimensions.

⇒ Let $f: U \to V$

&

$g: U \to V$ where U & V are manifolds.

⇒ A homotopy is a continuous function

$H: U \times [0,1] \rightarrow V$

Such that,

$H(x,0) = f(x)$

$H(x,1) = f(y)$

Example,

$f: R \rightarrow R \, , \, g: R \rightarrow R$

$H(x,t) = (1-t)f(x) + t g(x)$

⇒ If there exist such a continuous mapping that deforms f to g, then f & g are homotopic.

↳ Resulting equivalence relation is denoted $f \sim g$.

⇒ We can also say that two paths f and g are path-homotopic ($f \sim g$) if they continuously deform into one another.

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

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

{where c is a representative element of the class}

⇒ Let $A \subset X$ and let $f: X \to A$ be a retraction.

⇒ A <u>deformation retraction</u> is a homotopy $H: X \times [0,1] \to X$ such that

- $H(x,0) = x$
- $H(x,1) \in A$
- $H(a,t) = a \quad \forall\, a \in A \;\&\; t \in [0,1]$

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

⇒ This retract is called <u>deformation retract</u>.

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

↳ A point y in a neighbourhood of x also continuously moves through X to a point in A such that $H(x,t)$ and $H(y,t)$ are close to each other as t varies from 0, to 1.

⇒ Diffeomorphism
↳ Preserves the structure of two spaces of same dimension.

Deformation retraction
↳ Preserves the structure of two spaces of different dimension.

⇒ Key topological properties of deformation retracts

① → It preserves 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.
↳ is denoted by $\pi_1(X, x_0)$ for loops in X passing through x_0.
↳ Since this is a group, it has a group operator $(*)$ that simply concatenates path.

⇒ A set X is simply connected if the fundamental group associated with the set $\pi_1(X, x_0)$ contains only the identity element.

⇒ 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.

② Deformation retracts have the properties of <u>connectivity</u>, <u>accessibility</u> & <u>departability</u>.

⇒ The GVD is a retract because the RM has been shown to be continuous and maps all points on the GVD to the GVD.

⇒ The GVD is a deformation retract because RM has been shown to be homotropic to the identity map.

★ <u>GVD ~~Diagram~~ Dimension</u>

<u>(The Preimage Theorem & Critical Points)</u>

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

⇒ In a plane, the GVD consists of one dimensional manifolds.

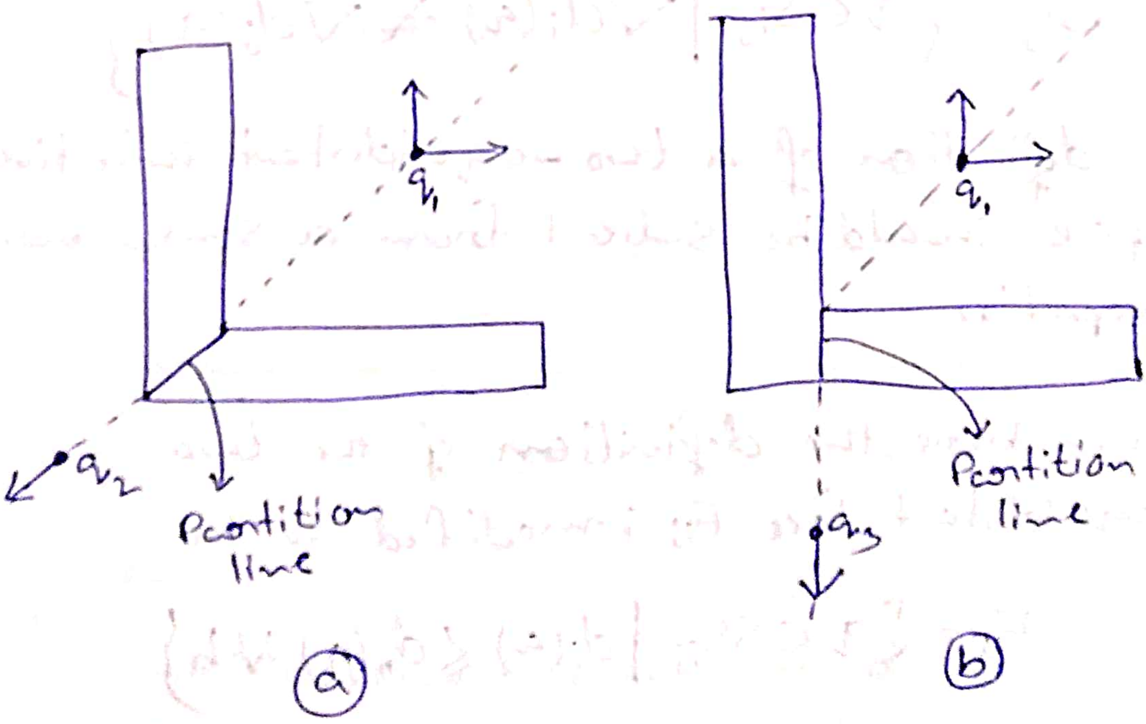
⇒ We are using the distance function d_i to define the GVD, but this function assumes that the obstacles are convex.

↳ This is unrealistic in most situation.

⇒ 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 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.



⇒ Note

↳ For the portions between the two arms, the gradients to the closest obstacles are distinct.

$$\nabla d_i(q_1) = \nabla d_j(q_1)$$

↳ For the other portions the gradients line up

$$\nabla d_i(q_2) = \nabla d_j(q_2)$$
$$\nabla d_i(q_3) = \nabla d_j(q_3)$$

⇒ Eliminating the portion of the two-equidistant surface with nondistinct gradient vectors yields a set termed the two-equidistant surjective surface denoted as:

$$SS_{ij} = \{ q \in S_{ij} \mid \nabla d_i(q) \neq \nabla d_j(q) \}$$

⇒ This definition of a two-equidistant surjective surface should be salient from a sensor-based perspective.

⇒ From here, the definition of the two-equidistant face F_{ij} is modified to be

$$F_{ij} = \{ q \in SS_{ij} \mid d_i(q) \leq d_h(q) \ \forall h \}$$

$\Rightarrow d_i(q) = d_j(q)$

$\Rightarrow d_i(q) - d_j(q) = 0$

$\Rightarrow (d_i - d_j)(q) = 0$

$\Rightarrow$ In other words, equidistance is the preimage of zero under the map

$$(d_i - d_j) : Q \to R$$

Theorem 5.2.2 (Preimage Theorem)

Let M and N be manifolds. Let $G : M \to N \in C^\infty$ and $n \in N$ be a regular value of G.

The set $G^{-1}(n) = \{m \in M \mid G(m) = n\}$ is a closed submanifold of M with tangent space given by

$$\boxed{T_m G^{-1}(n) = \operatorname{Ker} DG(m)}$$

If N is finitely dimensional, then

$$\dim(G^{-1}(n)) = \dim(M) - \dim(N)$$

{ Kernel or null space }

$$\Downarrow$$

$$\boxed{\dim(G^{-1}(n)) = \dim(M) - \dim(N)}$$

Note

$\to T_m M$ is tangent space at m on the manifold

~~→ The differential DG(m) is surjective~~

$\Rightarrow$ A regular value is an n where for all $m \in G^{-1}(n)$, the differential $DG(m)$ is surjective.

⇒ A critical point is a point where the differentiation is not surjective and hence loses rank.

⇒ Let $\Sigma(G)$ be the set of all critical points

⇒ For all $q^* \in \Sigma(G)$, $G(q^*)$ are critical values.

⇒ All points $q \notin \Sigma(G)$ where $DG(q)$ is surjective are termed regular points with $G(q)$ as their corresponding regular values.

⇒ So, by the preimage theorem SS_{ij} is one-dimensional in the plane.

★ Construction of the GVD

⇒ We discuss three method for constructing the planar GVD:

ⓐ① Using sensor information allowing the robot to construct the GVD in a unknown space.

ⓑ② Assuming the world has polygon obstacles in which case we can compute complexity information about the GVD.

ⓒ③ Assuming the world is a grid allowing for efficient computation.

(a) Sensor-based Construction of the GVD

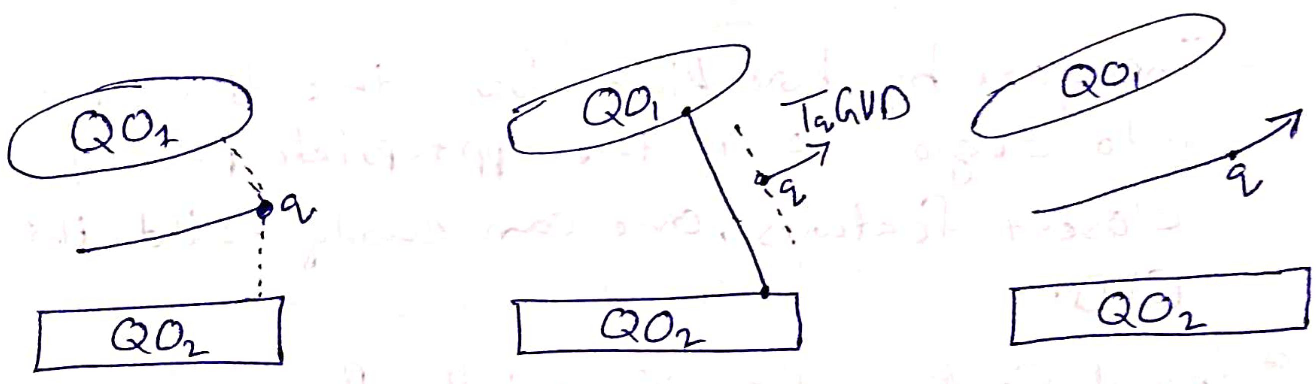
⇒ The GVD can be incrementally constructed because it is defined in terms of distance information

↳ which is readily provided by range sensor onboard mobile robot.

(i) Using line of sight data, the robot initially accesses the GVD.

↳ By simply moving away from the nearest obstacle until it is equidistant to two obstacles.

(ii) Then begin tracing an edge until it encounters a meet point or a boundary point.



⇒ The target is the null space of $\nabla G(q)$, which corresponds to a line orthogonal to $\nabla d_i(q) - \nabla d_j(q)$.

$$\boxed{G(q) = d_i(q) - d_j(q)}$$

{ This is identical to passing a line through the two closest points and taking the vector perpendicular to the line to be the target }

⇒ A meet point is detected by looking for a sudden change in one of the two closest obstacle.

(iii) When all meet points have no unexplored edges associated with them, exploration is complete.

ⓑ Polygonal Spaces

⇒ In a polygonal environment, obstacles have two features, vertices and edges, thereby making equidistance relationship easy to define.

⇒ The set of points equidistant to two vertices is a line.

⇒ The set of points equidistant to two edges is a line.

⇒ The set of points equidistant to a vertex and an edge is a parabola.

⇒ Therefore by breaking down the free space into regions with the appropriate pair of closest features, one can easily build the GVD.

ⓒ Grid Configuration Spaces: The Brushfire method

⇒ Input for the brushfire method:
- ↳ grid of zeros corresponding to free space.
- ↳ ones corresponding to an obstacle.

⇒ The output of the brushfire method is a discrete map where each pixel in the grid has a value equal to the distance to the closest point on the closest obstacle.

3. Retract-like Structures:
(The Generalized Voronoi Graph)

⇒ Now, we consider the case when $Q = \mathbb{R}^3$.
↳ In $\mathbb{R}^3$, the GVD is 2D and therefore reduces the motion planning problem by a single dimension.

⇒ Just as two planes in $\mathbb{R}^3$ generally intersect on a line, two two-equidistant faces intersects and form a one dimensional manifold.

⇒ The union of these one-dimensional structure is termed the generalized Voronoi graph (GVG).

★ GVG Dimension: ~~Traversality~~ Transversality

⇒ The GVG edges in $\mathbb{R}^3$ are the set of points equidistant to three obstacles such that the three obstacles are closest and have distinct gradients.

⇒ Let $S_{ijk} = \{ q \mid (d_i - d_j)(q) = 0 \;\&\; (d_i - d_k)(q) = 0 \}$

⇒ Just like before, we are interested in subset of S_{ijk} where the gradients are distincts.

$$SS_{ijk} = \{ q \in S_{ijk} \mid \nabla d_i(q) \neq \nabla d_j(q), \nabla d_i(q) \neq \nabla d_k(q), \nabla d_j(q) \neq \nabla d_k(q) \}$$

$$SS_{ijk} = SS_{ij} \cap SS_{ik} \cap SS_{jk}$$

⇒ To determine the dimension of the GVG edge, we look at $G: \mathbb{R}^3 \to \mathbb{R}^3$

where

$$G(a) = \begin{bmatrix} (d_i - d_j) \\ (d_i - d_k) \end{bmatrix} (a)$$

⇒ Preimage $G^{-1}(0)$ is the set of points equidistant to three obstacle QO_1, QO_2 and QO_3, when the differential $DG(a)$ is surjective (i.e. does not lose rank)

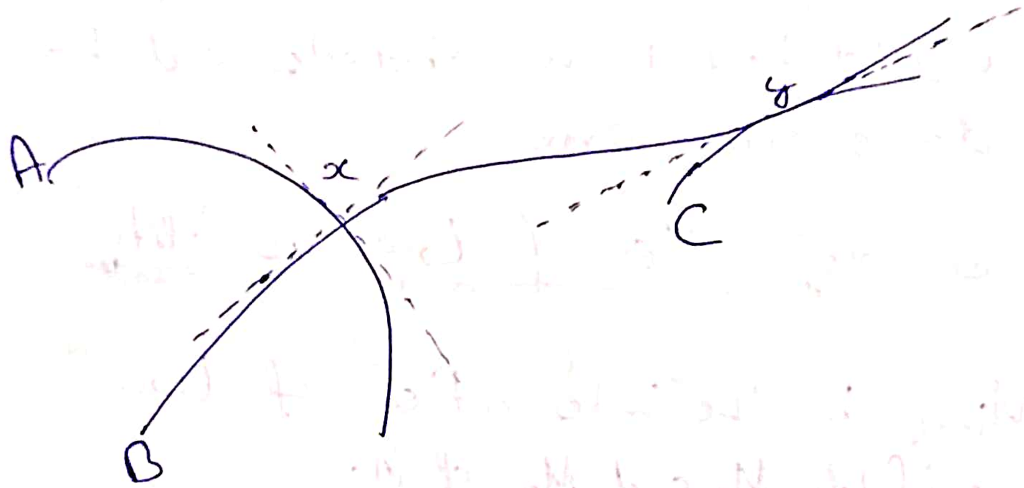
⇒ <u>Transversality, a property of how set intersect</u>

- Two intersecting lines in a plane.
  - → These lines may intersect in one of three ways.
    - → not at all (Parallel)
    - → At a point (generic)
    - → On a line (overlap)
  - → Parallel & overlap cases can be viewed as Unstable.
  - → The point intersection can be viewed as stable.
  - → we call:
    - → Stable ⇒ Transversal
    - → nonstable ⇒ non-Transversal

⇒ In actuality, transversality is a local property of manifold.

⇒ Since transversality is a local property, we look at the intersection of the tangent spaces, not of the manifolds themselves.

⇒ If intersection of the tangent spaces is transversal at a point, then the manifolds intersect transversally at that point.



⇒ We know from the preimage theorem that the tangent space $T_a G^{-1}(0) = \{ v \in T_a Q \mid DG(a) v = 0 \}$

⇒ We assume that surjective equidistance sheets intersect transversally at all points.

⇒ Therefore $DG(a)$ has full rank and we can use the preimage theorem to assure us that SS_{ijk} is indeed a Om-dimensional manifold.

⇒ The GVG in R^3 is then the Union of

$$F_{ijk} = \{ q \in SS_{ijk} \mid d_i(q) \leq d_h(q) \ \forall h \}$$

$$GVG = \bigcup_i \bigcup_j \bigcup_k F_{ijk}$$

⇒ In higher dimensions one can define more equidistant sheets and intersect them to form a GVG.

↳ In R^m, the GVG is the set of points equidistant to m obstacles and has dimension one.

⇒ <u>Formal defination of transversality</u>

⇒ Let M_{int} be the intersection of two submanifolds M_1 and M_2 of M.

↳ The intersection is said to be transversal if

$$T_x M_1 + T_x M_2 = T_x M. \ \forall \text{ points } x \in M_{int}.$$

⇒ If M_1 & M_2 are finitely dimensional,

↳ transversality implies that,

$$\boxed{\begin{gathered} \operatorname{Codim}(T_x M_1 \cap T_x M_2) = \operatorname{Codim}(T_x M_1) + \operatorname{Codim}(T_x M_2) \\ \forall x \in M_{int} \end{gathered}}$$

★ Retract-like Structure Connectivity

⇒ GVG is typically not connected and thus is not a roadmap.

⇒ The lack of connectivity of the GVG is not the fault of the GVG definition, but rather a consequence of using deformation retractions.

↳ In general, there cannot be a one-dimensional deformation retract of a punctured 3 or more dimensional space.

⇒ We address the lack of connectivity of the 1D structure by first looking at a connected 2D structure, and then defining 1D structure on the two-dimensional structure to form a roadmap.

⇒ We can exploit connectivity of the GVD to "patch together" the GVG.

$$F_{ijk} = \delta F_{ij} \cap \delta F_{ik} \cap \delta F_{jk}$$

⇒ If & only if, the boundaries of all two-equidistant sheets were connected, then the resulting GVG would be connected.

⇒ So, our goal now is to connect the boundaries of each of the two equidistant sheets.

⇒ To connect the GVG edges, we define additional structures called higher-order GVG edges

⇒ A second order GVG edge $F_{kl}|_{F_{ij}}$ is the set of points where QO_i and QO_j are the closest pair of equidistant obstacles and QO_k and QO_l are the second closest.

$$F_{kl}|_{F_{ij}} = \{ q \mid d_i(q) = d_j(q) < d_k(q) = d_l(q) \leq d_h(q) \ \forall h \neq i, j, k, l \text{ s.t. } \nabla d_i(q) = \nabla d_j(q) \text{ and } \nabla d_k(q) \neq \nabla d_l(q) \}$$

⇒ The second-order GVG edge are essentially planar-GVG edges but defined on two equidistant faces.

⇒ The preimage theorem guarantees that these edges are one-dimentional and terminate (and intersect) at second order meet points, denoted as $F_{klp}|_{F_{ij}}$

⇒ We call the union of GVG and second-order GVG the hierarchical generalized Voronoi graph (HGVG).

↳ By itself is not connected

⇒ However, there is a clue in the second-order GVG that directs the planner to look for a separate GVG-connected component.

⇒ Period ⇒ Closed-loop path in second-order GVG

⇒ Once a period is detected, the planner can trace a path that maintains two-way equidistance between QO_i and QO_j while decreasing the distance to QO_K.

⇒ Such a path follows, in general, the negative gradient of d_K because we start with $d_K(q) > d_i(q) = d_j(q)$, and decreasing d_K yields a configuration where $d_K(q) = d_i(q) = d_j(q)$.

⇒ However in order to maintain double equidistance between QO_i and QO_j the negative gradient must be projected onto the two-equidistant face.

↳ Hence the path $\dot{c}(t) = -\pi_{T_{c(t)}F_{ij}} \nabla d_K(c(t))$

⇒ Following the projected negative gradient traces a path that terminates on a GVG where $d_i(q) = d_j(q) = d_K(q)$ as long as $\pi_{T_{c(t)}F_{ij}} \nabla d_K(c(t))$ does not vanish.

↳ If it vanishes, then no such GVG edge exist in which case the robot returns to the second-order period to continue exploration.

⇒ This is just the beginning of what is required for connectivity.

↳ Ensuring connectivity can be quite tedious & challenging.

★ **Lyapunov Control: Sensor-Based Construction of the HGVG**

⇒ Exploration with the HGVG shares the same key steps as GVD exploration:

① Access the HGVG
② Explicitly "trace" the HGVG edges
③ Determine the location of nodes
④ Explore the branches emanating from the nodes.
⑤ Determine when to terminate the terminate the tracing procedure.

⇒ Accessing the GVG is a sequence of gradient ascent operations.

↳ The robot moves away from the closest obstacle until it is two-way equidistant.

↳ Then, while maintaining two-way equidistance, the robot increases distance until it is three-way equidistant

- **GVG Edge Tracing**

⇒ Let $G(q) = \begin{bmatrix} d_i - d_j \\ d_i - d_k \end{bmatrix}(q) = 0$

↳ Robot is equidistant to three obstacles on the GVG.

⇒ At a point q in the neighborhood of the interior of a GVG edge, the robot steps in the direction.

$$\boxed{\dot{q} = \alpha V + \beta (DG(q))^{+} G(q)}$$

Where,

→ α and β are scalar gain

→ $V \in Null(DG(q))$

→ $(DG(q))^{+}$ is the Penrose pseudoinverse.

⇒ $DG(q)$ can be solely computed from range sensor information.

⇒ When q is on the GVG, $G(q) = 0$.

$$\dot{q} = \alpha V$$

⇒ When q is not on the GVG, then

$$(DG(q))^{+} G(q) \neq 0$$

↳ {This term corresponds to the correction step}

α ⇒ Determines how quickly the robot moves along the GVG

β ⇒ Determines how ~~quickly the robot~~ aggressively the robot moves back to the GVG.

⇒ To determine stability of the control law let $\Gamma = \frac{1}{2} G^T G$ measures the distance a point q is away from the GVG

$$\Gamma(q) = \frac{1}{2} G^T(q) G(q)$$

$$\dot{\Gamma}(q) = G^T(q) \dot{G}(q)$$

$$= G^T(q)\, DG(q)\, \dot{q}$$

$$= G^T(q)\, DG(q) \left(\alpha\, NULL(DG(q)) + \beta \left(DG^+(q) G(q)\right)\right)$$

$$= \beta\, G^T(q)\, DG(q)\, DG^+(q)\, G(q)$$

$$= \beta\, G^T(q) G(q)$$

⇒ if $\beta < 0$, then $\dot{\Gamma}$ is negative.

⇒ This assures that Γ decreases to zero, meaning that equation direct the robot onto the GVG.

- Meet Point Homing

⇒ While generating the GVG, the robot must precisely locate itself on the meet points.

↳ A meet point homing algorithm can be used to stably converge onto the meet point location.

⇒ The control law for homing onto a meet point is similar to the one for generating GVG edge, except G is now defined as:

$$G(q) = \begin{bmatrix} d_1 - d_2 \\ d_1 - d_3 \\ d_1 - d_4 \end{bmatrix}(q) = 0$$

⇒ Since $Null(DG(q)) = 0$, the controller is

$$\dot{q} = \beta (DG(q))^{+} G(q)$$

- Higher-Order GVG Control Laws

⇒ Naturally, by varying the G function, one can trace different structures.

⇒ A second order GVG edge has

$$G(q) = [d_i - d_j, d_k - d_l]^T.$$

↳ Likewise, a second order meet point has

$$G(q) = [d_i - d_j, d_k - d_l, d_k - d_p]^T(q)$$

4 Piecewise Retracts

(The Rod-Hierarchical Generalized Voronoi Graph)

⇒ The previous roadmaps were defined for a point robot in a workspace which has a Euclidean configuration space.

⇒ Even when full knowledge of the workspace is available prior to the planning event, constructing non-Euclidean configuration space can be quite challenging.

⇒ In this section, we define a roadmap for a line segment operating in the plane.

↳ Sometime we call this line segment a <u>rod</u>.

⇒ To distinguish among previous roadmaps

↳ Let the point-GVG & point-HGVG be structures defined for a point robot in a Euclidean configuration space.

⇒ Configuration space for the rod is SE(2).

↳ This is 3D.

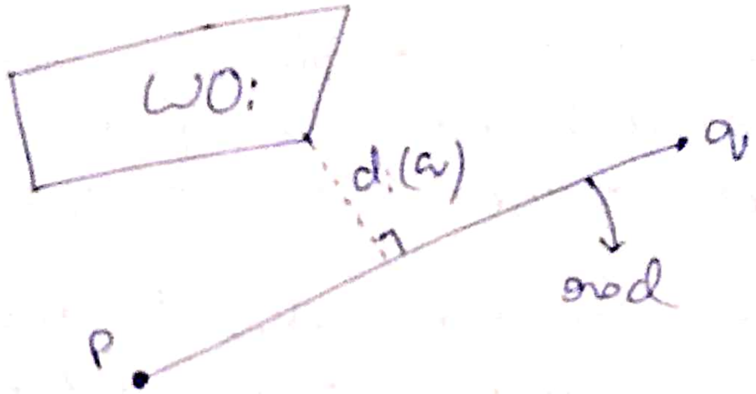
↳ So it makes sense to look at the set of configurations equidistant to three obstacles.

⇒ We measure distance in the workspace, not configuration space.

⇒ Let $R(q) \in W$ be the set of points the rod occupies in the workspace when it is at configuration q.

⇒ Let d_i be distance to obstacle WO_i for the rod robot.

$$d_i = \min_{r \in R(q),\ c \in WO_i} d(r, c)$$



⇒ Using this ~~rotat~~ notion of distance, a rod-two equidistant face

$$\boxed{RF_{ij} = \{ q \in SE(2) \mid d_i(q) - d_j(q) = 0 \ \& \ \nabla d_i(q) \neq \nabla d_j(q) \}}$$

↳ Then the rod-GVG edge is

$$\boxed{RF_{ijk} = RF_{ij} \cap RF_{ik} \cap RF_{jk}}$$

⇒ Just like the point-GVG in $\mathbb{R}^3$, the collection of rod-GVG edges does not necessarily form a connected set.

↳ To produce a connected structure we introduce another type of edge, called an R-edge.

↳ {Set of rod configurations tangent to the planar point-GVG}

→ The ord-HGVG is a piecewise retract because it is formed by the union of deformation retracts of subsets of the configuration space.

↳ which are then linked together with the R-edges to form a connected roadmap.

5> Silhouette Methods

→ In contrast to looking at equidistance, the silhouette approaches use extrema of a function defined on a codimension one hyperplane called slice.

↳ {denoted by Q_λ}

→ λ parameterizes the slice.

↳ Varying the parameter λ has the effect of sweeping the slice through the configuration space.

↳ As the slice is swept through the configuration space, for each value of λ, the critical points of a function restricted to the slice are determined.

→ The trace of the critical points as the slice is swept through the configuration space does not necessarily form a connected set.

⇒ Therefore, the Silhouette methods look for another type of Critical point and then recursively call the algorithm on a slice passing through these Critical points.

↳ The resulting network of extremal Points forms the roadmap.

★ <u>Canny's Roadmap Algorithm</u>

⇒ Canny's work established fundamental Complexity bounds using roadmap theory.

⇒ For an environment populated by obstacles whose boundaries can be represented as <u>p polynomials</u> of <u>maximum degree ω</u> for some positive ω in Configuration space.

↳ Any navigation path-planning problem can be solved in $p^n(\log p)\omega^{(O(n^4))}$ time using his roadmap algorithm.

{n ⇒ dimension of Configuration space}
or
dof of the robot

⇒ In this method, the choice of initial sweep direction is arbitrary, but for the sake of discussion, let's chose the q_1-direction.

⇒ As the slice is swept in the q_1-direction, "extremal points" in the q_2-direction are determined in each slice.

↳ The extremal points in the q_2-direction are extrema of the projection function $\pi_2 : \mathbb{R}^{m-1} \to \mathbb{R}$ where $\pi_2(q) = q_2$.

⇒ The extremal points of π_2 for all of the slices are the <u>silhouette curves</u>.

⇒ In general, the silhouette curves are not guaranteed to be connected, and hence may not form a roadmap.

↳ However, we can look at the slices where the number of silhouette curve changes.

↳ These slices are called <u>critical slices</u> & the λ values that parameterize critical slices are <u>critical values</u>.

⇒ The points on the silhouette curves where the silhouette curves are tangent to the critical slices are termed <u>critical points</u>.

⇒ On the critical slices, the silhouette algorithm is recursively invoked where the new swept slice now has one less dimension than the critical slice.

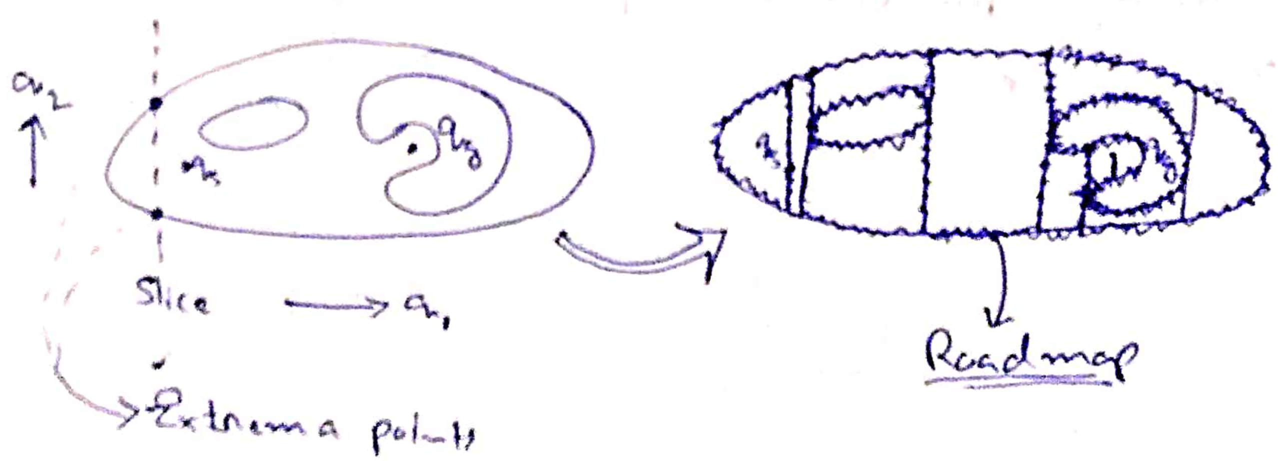
↳ The new silhouette comprises the trace of extremal points in the q_3 direction.

⇒ These silhouette curves may not be connected either, so this procedure is recursively invoked on lower-dimensional critical slices until there is no more critical points on the slice has one dimension.

⇒ In the latter case, the 1D slice is the silhouette.

⇒ Finally the union of the resulting silhouette curves forms the roadmap.

⇒ Accessibility & departability of the roadmap are achieved by treating the slices that contain q_{start} & q_{goal} as critical $(m-1)$-dimensional slices of the initial sweep.



★ Critical Points and Morse Function

⇒ In this Section, we define the Silhouette Curves in terms of Critical points of a function.

↳ The function has to be Morse.

⇒ The Slices themselves are also defined in terms of a function.

⇒ Originally, Canny suggested that a slice be the preimage of the projection operator π_1.

{π_1 Projects a point onto its first Coordinate

$$\pi_1(a) = a_1$$ }

⇒ We denote a slice as

$$\boxed{Q_\lambda = \{x \in Q \mid \pi_1(a) = \lambda\}} \text{ where } \lambda = a_1 \in R$$

$$U_\lambda Q_\lambda = Q$$

⇒ On each Q_λ, we look for extrema of π_2

{i.e. we look for extrema of $\pi_2|Q_\lambda$, where $\pi_2|Q_\lambda$ is projection operator restricted to the slice}

Lemma 5.5.1 (Lagrange Multiplier)

→ Let S be an n-manifold in $\mathbb{R}^{n+1}$, $S = f^{-1}(c)$ where $f: \mathbb{R}^{n+1} \to \mathbb{R}$ is such that $\nabla f(a) \neq 0 \ \forall a \in S$.

→ Suppose $h: \mathbb{R}^{n+1} \to \mathbb{R}$ is a smooth function and $P \in S$ is an extremal point of h on S.

→ Then there exists a real number μ such that $\nabla h(P) = \mu \nabla f(P)$ (the number μ is called the Lagrange multiplier)

→ In other words $\nabla f(P)$ is parallel to $\nabla h(P)$ at an extremum P of h on S.

Lemma 5.5.2 (Generalized Lagrange Multiplier)

Let M be the preimage of $f: \mathbb{R}^m \to \mathbb{R}^p$ & $h: \mathbb{R}^m \to \mathbb{R}^q$. The point x is a critical point of $h|_M$ if & only if the following matrix loses its rank.

$$D(f,h)_a \begin{bmatrix} \frac{\delta f_1}{\delta a_1}(a) & \cdots & \frac{\delta f_1}{\delta a_m}(a) \\ \vdots & & \vdots \\ \frac{\delta f_m}{\delta a_1}(a) & \cdots & \frac{\delta f_m}{\delta a_m}(a) \\ \frac{\delta h_1}{\delta a_1}(a) & \cdots & \frac{\delta h_1}{\delta a_m}(a) \\ \vdots & & \vdots \\ \frac{\delta h_q}{\delta a_1}(a) & \cdots & \frac{\delta h_p}{\delta a_m}(a) \end{bmatrix}$$

$$\boxed{\Sigma(\pi_{12}|s|) = \bigcup_{\lambda} \Sigma(\pi_2|\pi_1^{-1}(\lambda))}$$ {Slice lemma by Canny}

★ Opportunistic Path Planner

⇒ The opportunistic path planner (OPP) generalizes Canny's original roadmap algorithm by tracing the local maxima of any potential function that is Morse on a flat slice as the slice is swept through the configuration space.

⇒ Canny and Lin suggest that the distance function D evaluated on the slice be used as the potential function.

↳ Local maxima on the slice of the distance function are points on the OPP roadmap.

⇒ The trace of the local maxima as the slice is swept through the workspace or configuration space are termed <u>freeways</u>.

⇒ The algorithm works as follows:

- ↳ First, a fixed slice direction is chosen
- ↳ The algorithm initially traces a path from start to the roadmap by performing gradient ascent on the distance function in the slice that contains the start

→ Likewise, a path is traced from the goal to the freeway via slice-constrained gradient ascent.

→ These two actions correspond to
  → accessibility
  → departability

⇒ If the start & goal freeways are connected, then the algorithm terminates.

⇒ In general, the set of freeways will not be connected, and paths between neighboring freeways must be found.

→ The OPP freeways are connected via <u>bridge curves</u>.

⇒ The union of bridge & freeway curves, sometimes termed as a <u>skeleton</u>, forms the one-dimensional roadmap: