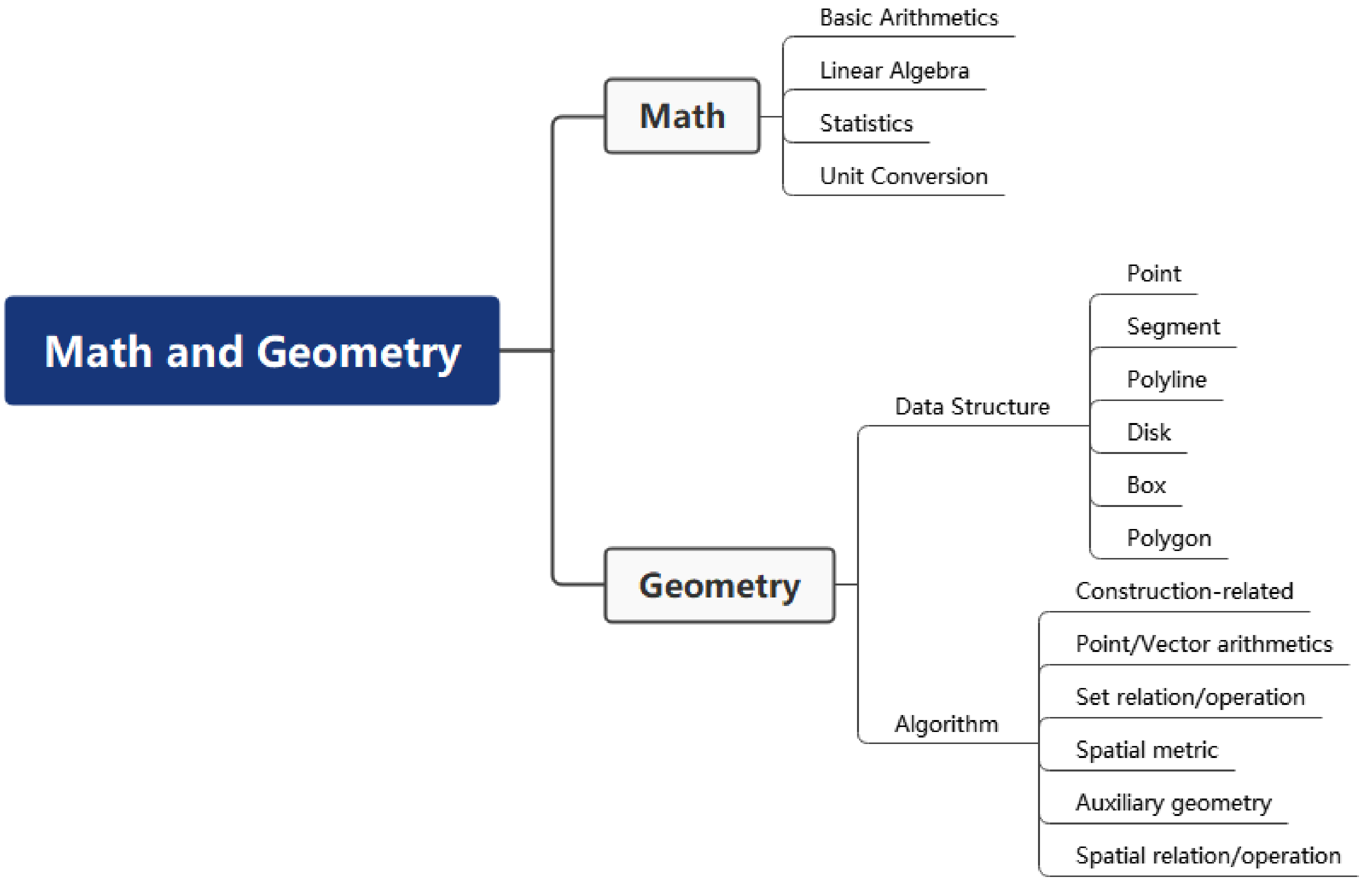
Application of geometry in path

CHH3213

Planning Control

CONTENTS

1 Math and Geometry Overview


Math and Geometry
Math
Basic Arithmetics
Linear Algebra
Statistics
Unit Conversion
Geometry
Data Structure
Point
Segment
Polyline
Disk
Box
Polygon
Algorithm
Construction-related
Point/Vector arithmetics
Set relation/operation
Spatial metric
Auxiliary geometry
Spatial relation/operation

Basic Knowledge of Geometry 2

Projection

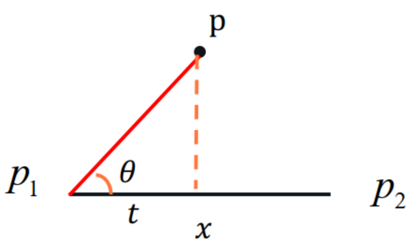
Find the projection point x of the point p on the segment p_1p_2.

$$x = p_1 + \frac{t}{\|p_1p_2\|} \cdot p_1p_2$$

$$t = p_1p \cdot \cos\theta$$

$$= \frac{p_1p \cdot p_1p_2}{\|p_1p_2\|}$$

$$\boxed{p_1 + p_1p_2 \cdot \frac{p_1p \cdot p_1p_2}{\|p_1p_2\|^2}}$$

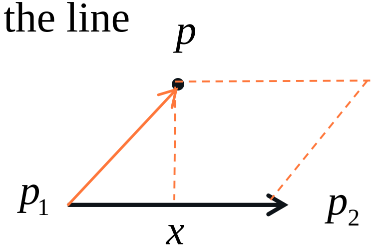


Distance

- Distance between two points

- Distance from the point to the line

$$d = \frac{p_1 p_2 \times p_1 p}{\left\| p_1 p_2 \right\|}$$

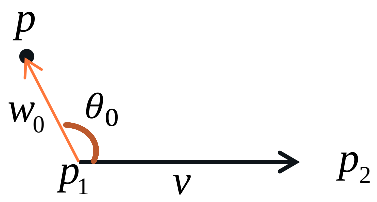


- Distance from point to segment

- Distance from segment to segment

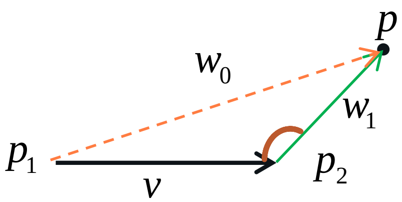
Distance

- Distance from point to segment



$$\theta_0 \in \left[-180^{\circ}, 180^{\circ}\right]$$

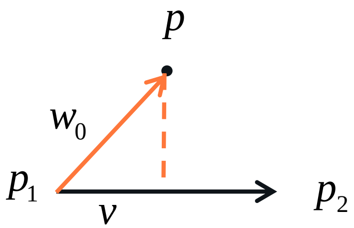
$$\mathbf{w}_0 \cdot \mathbf{v} \le 0 \Leftrightarrow |\theta_0| \ge 90^{\circ} \Leftrightarrow \mathrm{d}(P, \mathbf{S}) = \mathrm{d}\left(P, P_1\right)$$



$$\theta_1 \in \left[-180^{\circ}, 180^{\circ}\right]$$

$$\mathbf{w}_1 \cdot \mathbf{v} \ge 0 \Leftrightarrow \mathbf{w}_0 \cdot \mathbf{v} \ge \mathbf{v} \cdot \mathbf{v} \Leftrightarrow$$

$$|\theta_1| \le 90^{\circ} \Leftrightarrow \mathrm{d}(P, \mathbf{S}) = \mathrm{d}\left(P, P_2\right)$$

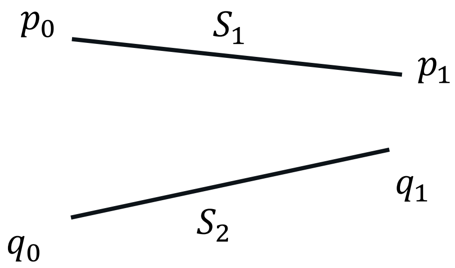


$$d(P, S) = \frac{v \times w_0}{\|v\|}$$

Distance

- Distance from segment to segment: Convert to the distance from point to segment.

$$\min\{d(p_0,S_2),d(p_1,S_2),d(q_0,S_1),d(q_1,S_1)\}$$



Distance

- Distance from line to line:

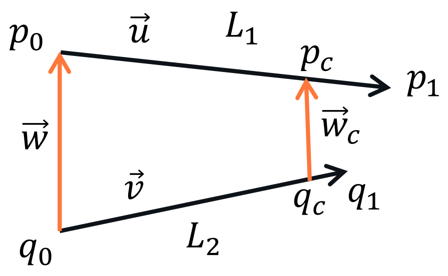
In any n-dimensional space:

$$\vec{u} = \vec{p}_1 - \vec{p}_0$$
$$\vec{v} = \vec{q}_1 - \vec{q}_0$$
$$\vec{u} \cdot \vec{w}_c = 0$$
$$\vec{v} \cdot \vec{w}_c = 0$$

$$\boxed{\vec{w}_c = \vec{w} + s\vec{u} - t\vec{v}}$$

$$0 = \vec{u} \cdot \vec{w} + s\vec{u} \cdot \vec{u} - t\vec{u} \cdot \vec{v}$$

$$0 = \vec{v} \cdot \vec{w} + s\vec{v} \cdot \vec{u} - t\vec{v} \cdot \vec{v}$$



$$d = \|\vec{w}_c\|$$

$$t = \frac{ae - bd}{ac - b^2}$$

$$s = \frac{be - cd}{ac - b^2}$$

$$a = \vec{u} \cdot \vec{u}$$
$$b = \vec{u} \cdot \vec{v}$$
$$c = \vec{v} \cdot \vec{v}$$
$$d = \vec{u} \cdot \vec{w}$$
$$e = \vec{v} \cdot \vec{w}$$

Whenever $ac - b^2 \neq 0$. when it's 0, the two lines are parallel.

Distance

- Distance from segment to segment:

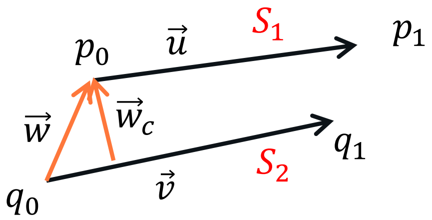
$$\boxed{\vec{w}_c = \vec{w} + s\vec{u} - t\vec{v}}$$

$$d = \|\vec{w}_c\|$$

$$\vec{u}\cdot\vec{w}_c = \vec{u}\cdot\vec{w} + s\vec{u}\cdot\vec{u} - t\vec{u}\cdot\vec{v}$$

$$\vec{v}\cdot\vec{w}_c = \vec{v}\cdot\vec{w} + s\vec{v}\cdot\vec{u} - t\vec{v}\cdot\vec{v}$$

- start of S_1 is closest to S_2

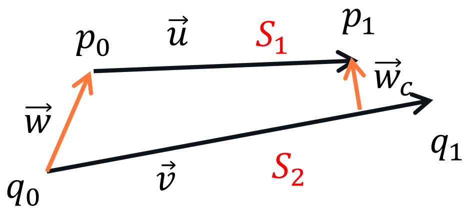


$$\vec{v}\cdot\vec{w}_c = 0$$
$$s = 0$$

$$0 = \vec{v}\cdot\vec{w} - t\vec{v}\cdot\vec{v}$$

$$t = \frac{\vec{v}\cdot\vec{w}}{\vec{v}\cdot\vec{v}} = \frac{e}{c}$$

- end of S_1 is closest to S_2



$$\vec{v}\cdot\vec{w}_c = 0$$
$$s = 1$$

$$0 = \vec{v}\cdot\vec{w} + \vec{v}\cdot\vec{u} - t\vec{v}\cdot\vec{v}$$

$$t = \frac{\vec{v}\cdot\vec{w} + \vec{v}\cdot\vec{u}}{\vec{v}\cdot\vec{v}} = \frac{e+b}{c}$$

Distance

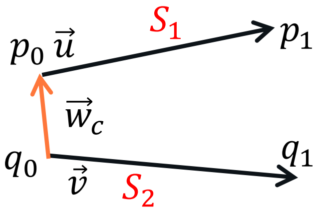
$$\boxed{\vec{w}_c = \vec{w} + s\vec{u} - t\vec{v}}$$

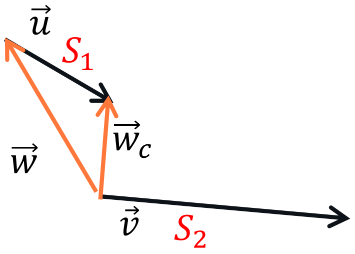
$$d = \|\vec{w}_c\|$$

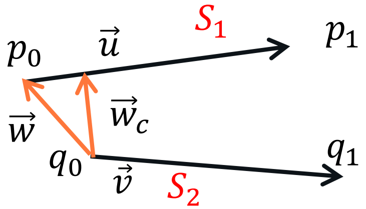
$$\vec{u}\cdot\vec{w}_c = \vec{u}\cdot\vec{w} + s\vec{u}\cdot\vec{u} - t\vec{u}\cdot\vec{v}$$

$$\vec{v}\cdot\vec{w}_c = \vec{v}\cdot\vec{w} + s\vec{v}\cdot\vec{u} - t\vec{v}\cdot\vec{v}$$

- Distance from segment to segment:
  - start of S_2 is closest to S_1







$s = 0$

$t = 0$

$$\vec{w}_c = \vec{w}$$

$s = 1$

$t = 0$

$$\vec{w}_c = \vec{w} + \vec{u}$$

$\vec{u}\cdot\vec{w}_c = 0$

$t = 0$

$$0 = \vec{u}\cdot\vec{w} + s\vec{u}\cdot\vec{u}$$

$$s = -\frac{\vec{u}\cdot\vec{w}}{\vec{u}\cdot\vec{u}} = -\frac{d}{a}$$

Distance

$$\boxed{\vec{w}_c = \vec{w} + s\vec{u} - t\vec{v}}$$

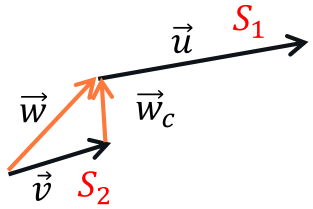
$$d = \|\vec{w}_c\|$$

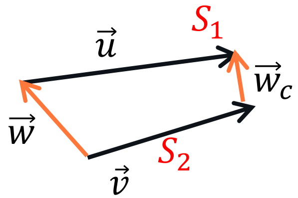
$$\vec{u}\cdot\vec{w}_c = \vec{u}\cdot\vec{w} + s\vec{u}\cdot\vec{u} - t\vec{u}\cdot\vec{v}$$

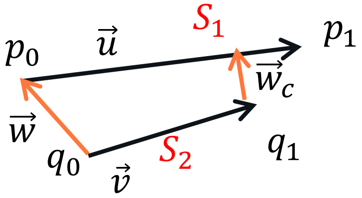
$$\vec{v}\cdot\vec{w}_c = \vec{v}\cdot\vec{w} + s\vec{v}\cdot\vec{u} - t\vec{v}\cdot\vec{v}$$

◆ Distance from segment to segment:

➢ end of S_2 is closest to S_1







$$s = 0$$
$$t = 1$$

$$\vec{w}_c = \vec{w} - \vec{v}$$

$$s = 1$$
$$t = 1$$

$$\vec{w}_c = \vec{w} + \vec{u} - \vec{v}$$

$$\vec{u}\cdot\vec{w}_c = 0$$
$$t = 1$$

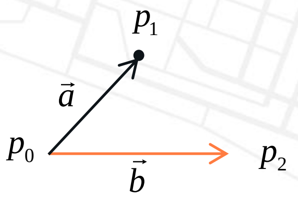
$$0 = \vec{u}\cdot\vec{w} + s\vec{u}\cdot\vec{u} - \vec{u}\cdot\vec{v}$$

$$s = \frac{-\vec{u}\cdot\vec{w} + \vec{v}\cdot\vec{u}}{\vec{u}\cdot\vec{u}} = \frac{-d+b}{a}$$

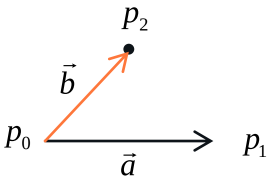
Determine which side of the segment the point is

The relative relationship between p0, p1, p2 can be generally described in the following ways：

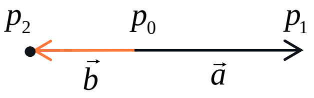
1. Clockwise $\vec{a} \times \vec{b} < 0$



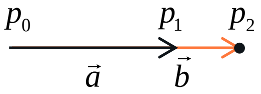
2. Counter-Clockwise $\vec{a} \times \vec{b} > 0$



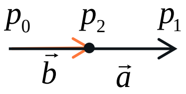
3. Online-Before $\vec{a} \times \vec{b} = 0 \,\&\, \vec{a} \cdot \vec{b} < 0$



4. Online-After $\vec{a} \times \vec{b} = 0 \,\&\, \vec{a} \cdot \vec{b} > 0 \,\&\, \|b\| > \|a\|$



5. WithIn $\vec{a} \times \vec{b} = 0 \,\&\, \vec{a} \cdot \vec{b} > 0 \,\&\, \|b\| < \|a\|$



Determine the position relationship by the cross product and dot product of $\vec{a}\ (p_1 - p_0)$ and $\vec{b}\ (p_2 - p_0)$.

Determine which side of two segments the point is

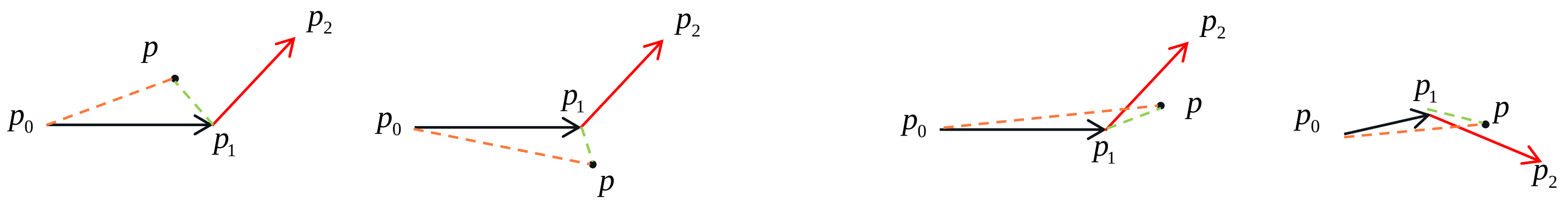
Finds polyline side using two segments starting at given index.

Judge $sign(p_0p \times p_0p_1)$ and $sign(p_1p_2 \times p_1p)$

(1) $p_0p_1 \times p_0p > 0 \,\&\, p_1p_2 \times p_1p > 0$ ➡ Left side

(2) $p_0p_1 \times p_0p < 0 \,\&\, p_1p_2 \times p_1p < 0$ ➡ Right side

$sign(p_0p_1 \times p_0p) \neq sign(p_1p_2 \times p_1p)$ ➡ Compare $\|p_0p_1 \times p_0p\|$ and $\|p_1p_2 \times p_1p\|$



(1) (2) (3)

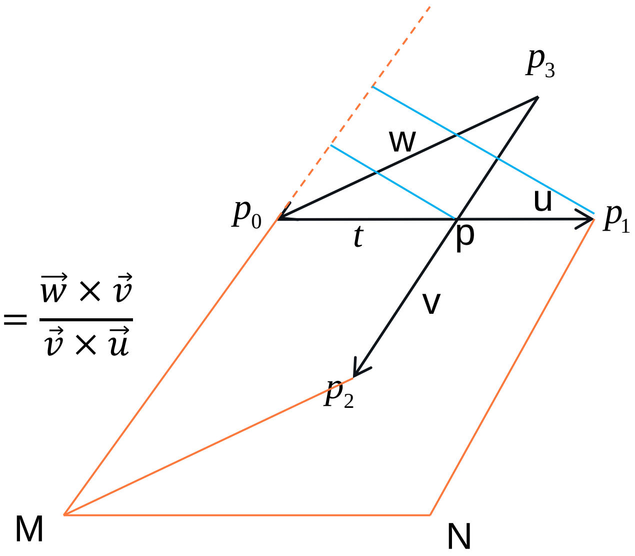
Intersection

Calculate the intersection between segment p_0p_1 and p_2p_3

$$\vec{v} = p_3p_2$$
$$\vec{u} = p_0p_1$$
$$\vec{w} = p_3p_0$$

$$t = \frac{\|p_0\mathrm{p}\|}{\|p_0p_1\|} = \frac{S_{p_3p_0Mp_2}}{S_{p_1p_0MN}} = \frac{\vec{w}\times\vec{v}}{\vec{v}\times\vec{u}}$$

$$p = p_0 + \mathrm{t}\cdot\vec{u}$$

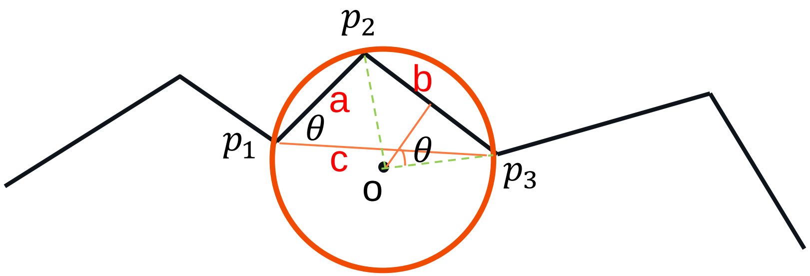


Curvature

$$\begin{cases} 2R\sin\theta = b \\ S = \dfrac{ac\sin\theta}{2} \\ \kappa = \dfrac{1}{R} \end{cases}$$

$$\kappa = \frac{4S}{abc}$$

$$\kappa = \frac{2p_1p_2 \times p_2p_3}{abc}$$

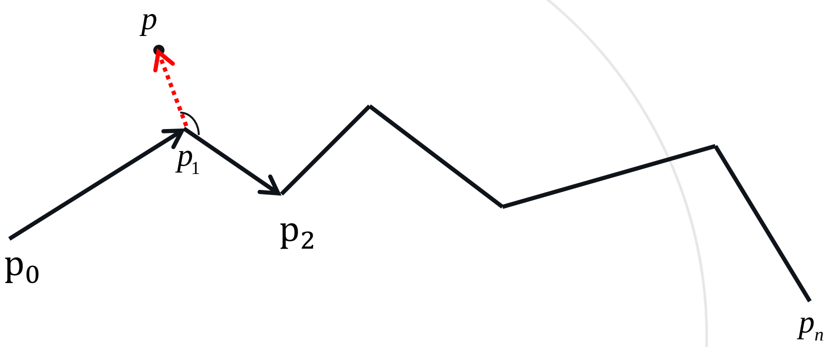


Geometric applications on path 3

Finds closest segment using linear search

Algorithm 1 Finds closest segment using linear search

Input: given point p.
Calculates the distance as minimum distance min_{dist} from given point p to the start point p_0
for $i = 1, \cdots, n$ **do**
  Compute Unit vector $\vec{v} = \frac{p_i - p_{i-1}}{\|p_i - p_{i-1}\|}$
  Compute $\vec{w} = p - p_{i-1}$
  Compute dot product $c_1 = \vec{v} \cdot \vec{w}$
  if $c_1 \leq 0$ **then**
    continue
  end if
  Compute $c_2 = \|p_i - p_{i-1}\|$
  if $c_2 \leq c_1$ **then**
    $dist = \|p - p_i\|$
  else
    compute projection point p_{pro} $dist = \|p - p_{pro}\|$
  end if
  Update minimum distance with $min_{dist} = \min\{min_{dist}, dist\}$
  Update closest segment index according to min_{dist}
end for
return closest segment index



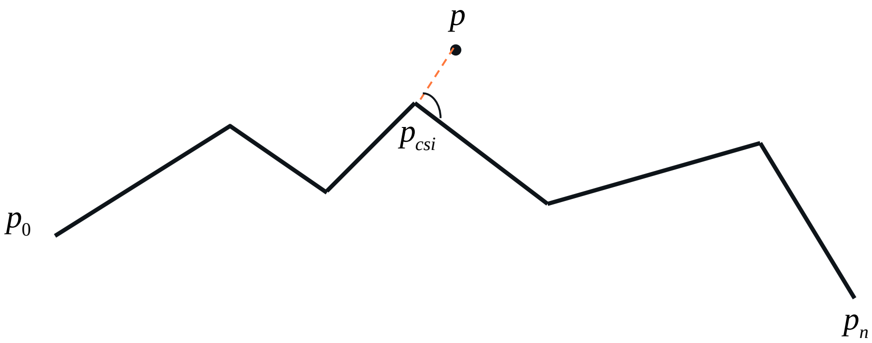
Find closest segment given hint

It’s similar to “finds closest segment using linear search”.

Since the reference index is given, we can search forward and backward from the reference index

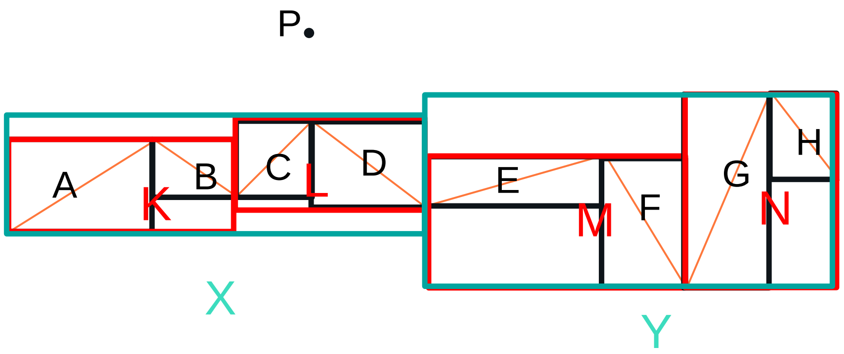
Algorithm 2 Finds closest segment given hint

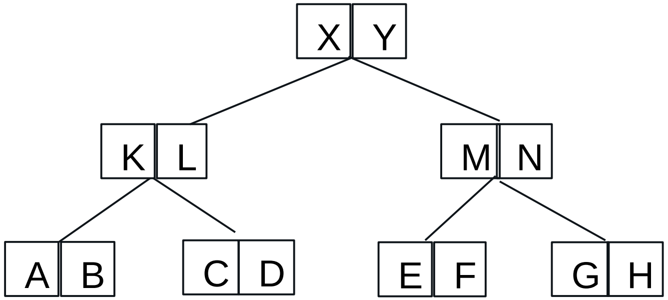
1: **Input**: given point p, given hint index csi.
2: Using linear search for closest segment in **forward** direction.
3: Using linear search for closest segment in **backward** direction.
4: **return** closest segment index.



Find closest segment using R-tree

Base implementation API： boost::geometry::index::rtree.
Linear search is suitable for cases with a small number of line segments.

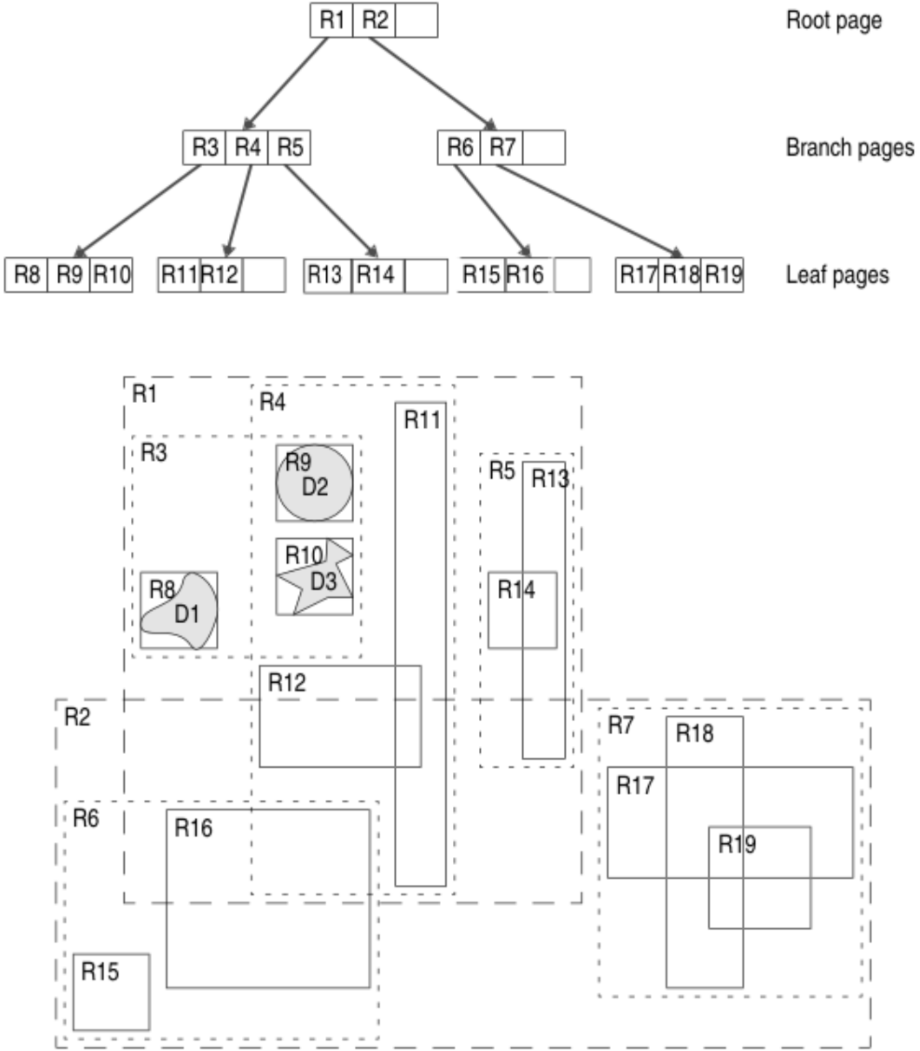




R-Tree

boost::geometry::index::rtree: This is self-balancing spatial index capable to store various types of Values and balancing algorithms.

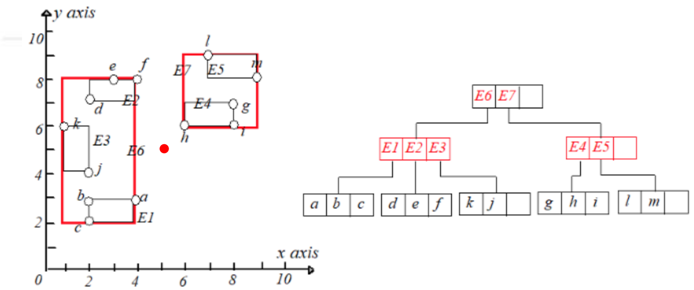
The implementation of R-tree is well encapsulated in boost::geometry, and the main thing we need to master is its querying skills.



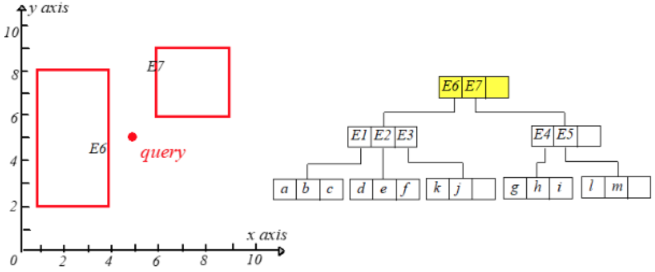
R-Tree

Nearest neighbours queries returns Values which are closest to some Geometry.

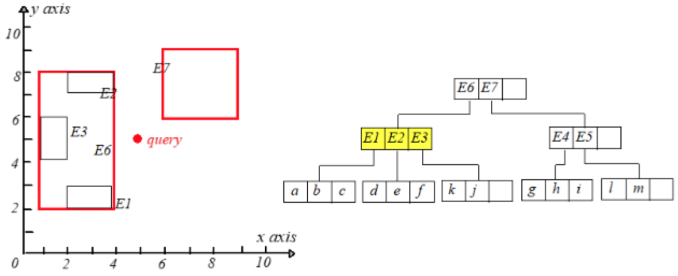
boost::geometry::index::nearest function uses Best-First Algorithm.



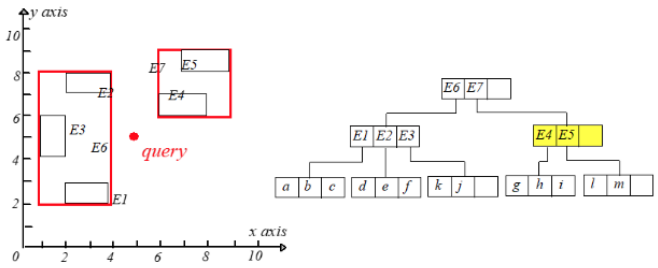
Action	Memory	Result
Visit Root	$E_6\sqrt{1}$, $E_7\sqrt{2}$	{empty}



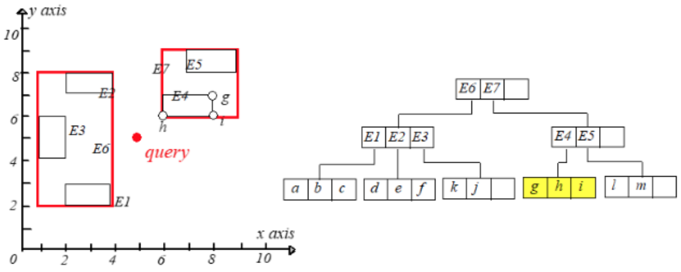
Action	Memory	Result
follow E_6	$E_7\sqrt{2}$, $E_1\sqrt{5}$, $E_2\sqrt{5}$, $E_3\sqrt{9}$	{empty}



Action	Memory	Result
follow E_7	$E_4\sqrt{2}$, $E_1\sqrt{5}$, $E_2\sqrt{5}$, $E_3\sqrt{9}$, $E_5\sqrt{13}$	{empty}



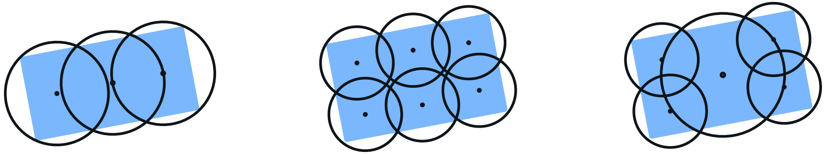
Action	Memory	Result
follow E_4	$h\sqrt{2}$, $E_1\sqrt{5}$, $E_2\sqrt{5}$, $E_3\sqrt{9}$, $i\sqrt{10}$, $E_5\sqrt{13}$, $g\sqrt{13}$	{empty}

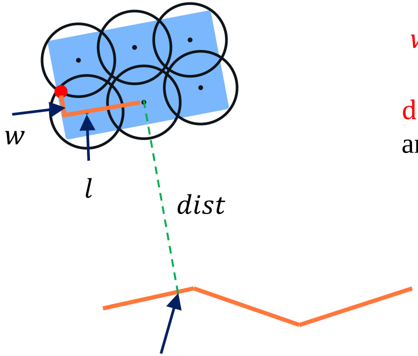


Addition Jacob computation

Computes the polyline distance given the a circle-based coverage of the vehicle.

circular disks to cover the actual shape of a vehicle.





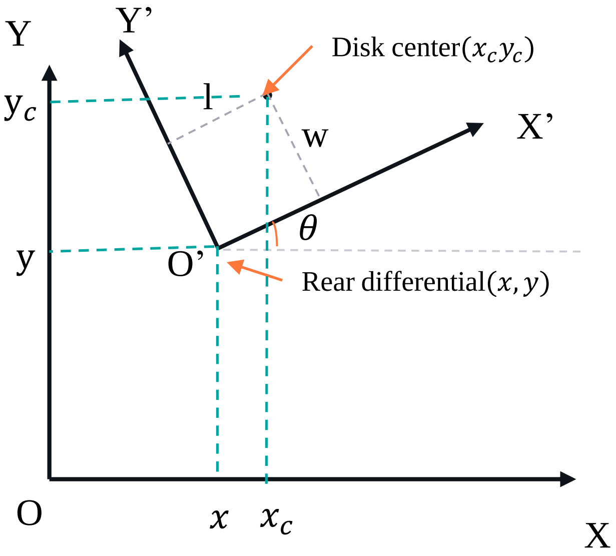
l : longitudinal offset from rear differential

w : lateral offset from rear differential

dist : the distance between the query point and the proximal point

Compute FullBody Polyline

How to compute jacabian?



$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} l \\ w \end{bmatrix}$$

$$x_c = x + l\cos\theta - w\sin\theta$$
$$y_c = y + l\sin\theta + w\cos\theta$$

$$\frac{\partial x_c}{\partial\theta} = -l\sin\theta - w\cos\theta$$

$$\frac{\partial y_c}{\partial\theta} = l\cos\theta - w\sin\theta$$

$$jacob_x = \frac{\partial dist}{\partial x} = \frac{\partial dist}{\partial x_c} \cdot \frac{\partial x_c}{\partial x} = \frac{\partial dist}{\partial x_c}$$
$$jacob_y = \frac{\partial dist}{\partial y} = \frac{\partial dist}{\partial y_c} \cdot \frac{\partial y_c}{\partial y} = \frac{\partial dist}{\partial y_c}$$
$$jacob_\theta = \frac{\partial dist}{\partial\theta} = \frac{\partial dist}{\partial x_c} \cdot \frac{\partial x_c}{\partial\theta} + \frac{\partial dist}{\partial y_c} \cdot \frac{\partial y_c}{\partial\theta}$$

https://github.com/CHH3213/Books/tree/master/%E6%95%B0%E5%AD%A6/%E8%AE%A1%E7%AE%97%E5%87%A0%E4%BD%95%E5%AD%A6

THANKS