main challenge: Space of all Robot controls (hence Trajectories)

are continuos, leading to a Complex Continuos-variable Optimization ladscape.
↳ Also, Designing a cost function that simulates Driving Behavior.
↳ Challenge: Paths that are Optimal but smooth and Avoiding Obst.

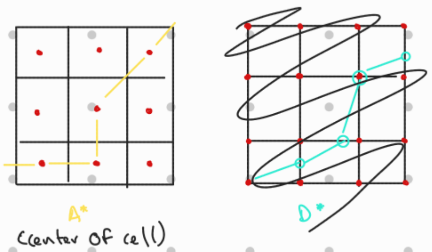
Steps:
1) Heuristic search in Continuos Coordinates for kinematic feasibility of Comp. Trajec.
2) Uses Conjugate Gradient (CG) descent to locally Improve The Quality.

* We need to produce a Driving Behavior that keep a Comfortable distance to obstacles.
↳ To penalize this, ppl use potential field Algorithm.
↳ Nuthless, in narrow passages it makes them untraversable. Thus:
3) Potential based on geometry of workspaces. Allowing nav. in narrow Passages.

## Hybrid-State A* Search

• Phase 1:

* A* algorithm applied to 3D kinematics State Space of vehicle
↳ But A modified state-Update rule that captures continuos State data in the discrete search nodes of A* (?)



A* (center of cell)

D*

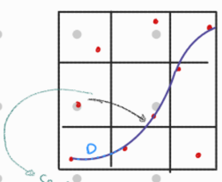
A* hybrid (A* hybrid-State Associates with each cell a Cont. 3D State of vehicle)

→ not the fastest, but DRIVABLE

• The Search Algorithm is based on two heuristics:

1) Ignores Obstacles but take into Account the non-holonomic nature of car
↳ Assume a goal state $(x_g, y_g, \Theta_g) = (\_,\_,\_)$ ↳ freedom Angles.
↳ Then we calculate the shortest path to the goal from every point $(x, y, \Theta)$ Assuming no Obstacles
↳ Se utiliza el max. del costo y la Distancia en 2D como heurística
↳ elimina Ramas que se Acercan Pero TRAEN ORIENT. INCORRECTAS

* No depende de los Sensores, es precomputada desde el Principio.



Se Acerca pero TRAE ORientación Incorrecta

→ * No hay obstáculos, PERO no te puedes mover como sea.

2) Ignores non-holonomic Car but uses Obstacle map by performing dynamic 2D Prog.
→ * Hay obstáculos Pero te puedes mover como sea.

How to Calculate the cost? we use Voronoi Field Path-cost funct.

↳ It is to Calc. the trade off btwn path length and prox. to obstacles.

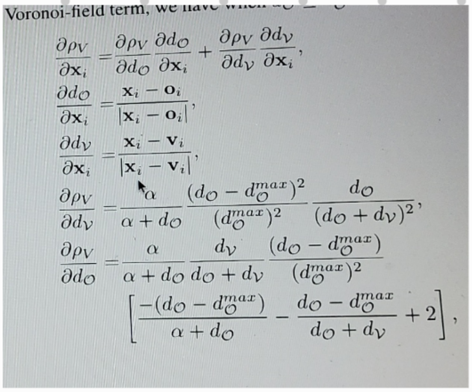
→ constant      → Distance to edge of GVD

$$\rho_v(x,y) = \left(\frac{\alpha}{\alpha + d_O(x,y)}\right)\left(\frac{d_v(x,y)}{d_O(x,y) + d_v(x,y)}\right)\left(\frac{(d_O - d_O^{max})^2}{(d_O^{max})^2}\right)$$

↳ Distance to Obstacle.

* Advantage =
  Field value is scale in proportions to the available paths for navigation.
  ↳ narrow spaces still drivable

* This produces Drivable paths but not smooth paths.
  ↳ For that



Voronoi-field term, we have

$$\frac{\partial \rho_V}{\partial \mathbf{x}_i} = \frac{\partial \rho_V}{\partial d_{\mathcal{O}}}\frac{\partial d_{\mathcal{O}}}{\partial \mathbf{x}_i} + \frac{\partial \rho_V}{\partial d_{\mathcal{V}}}\frac{\partial d_{\mathcal{V}}}{\partial \mathbf{x}_i},$$

$$\frac{\partial d_{\mathcal{O}}}{\partial \mathbf{x}_i} = \frac{\mathbf{x}_i - \mathbf{o}_i}{|\mathbf{x}_i - \mathbf{o}_i|},$$

$$\frac{\partial d_{\mathcal{V}}}{\partial \mathbf{x}_i} = \frac{\mathbf{x}_i - \mathbf{v}_i}{|\mathbf{x}_i - \mathbf{v}_i|},$$

$$\frac{\partial \rho_V}{\partial d_{\mathcal{V}}} = \frac{\alpha}{\alpha + d_{\mathcal{O}}}\frac{(d_{\mathcal{O}} - d_{\mathcal{O}}^{max})^2}{(d_{\mathcal{O}}^{max})^2}\frac{d_{\mathcal{O}}}{(d_{\mathcal{O}} + d_{\mathcal{V}})^2},$$

$$\frac{\partial \rho_V}{\partial d_{\mathcal{O}}} = \frac{\alpha}{\alpha + d_{\mathcal{O}}}\frac{d_{\mathcal{V}}}{d_{\mathcal{O}} + d_{\mathcal{V}}}\frac{(d_{\mathcal{O}} - d_{\mathcal{O}}^{max})}{(d_{\mathcal{O}}^{max})^2}\left[\frac{-(d_{\mathcal{O}} - d_{\mathcal{O}}^{max})}{\alpha + d_{\mathcal{O}}} - \frac{d_{\mathcal{O}} - d_{\mathcal{O}}^{max}}{d_{\mathcal{O}} + d_{\mathcal{V}}} + 2\right],$$

ure 7: Hybrid-A* and CG paths for a complicate