Motion planning for hundreds of floating robots

Jan Kamm*, Antonio Terpin*, Raffaello D'Andrea, Aswin Ramachandran

ETH Zürich · Institute for Dynamic Systems and Control
* Equal contribution



ETH zürich

IDSC

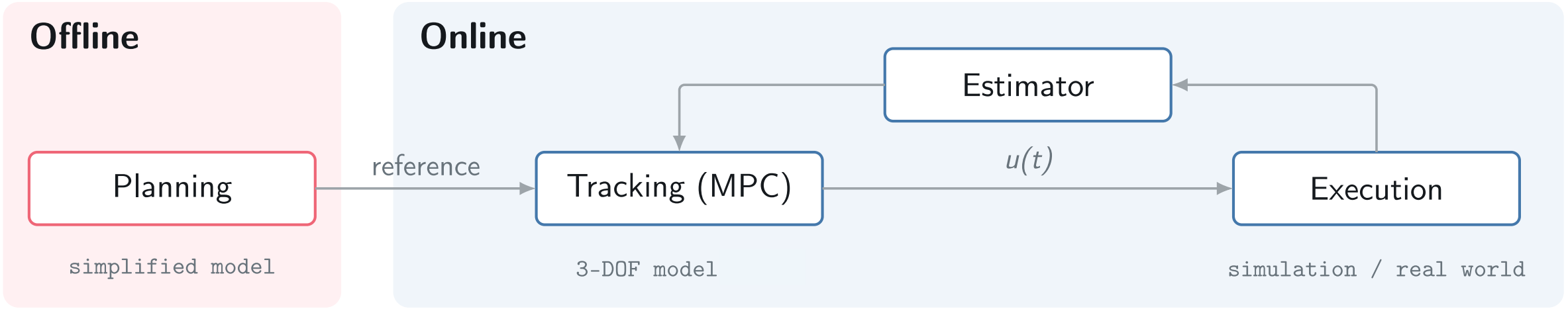
WAY of WATER

IROS 2026
PITTSBURGH

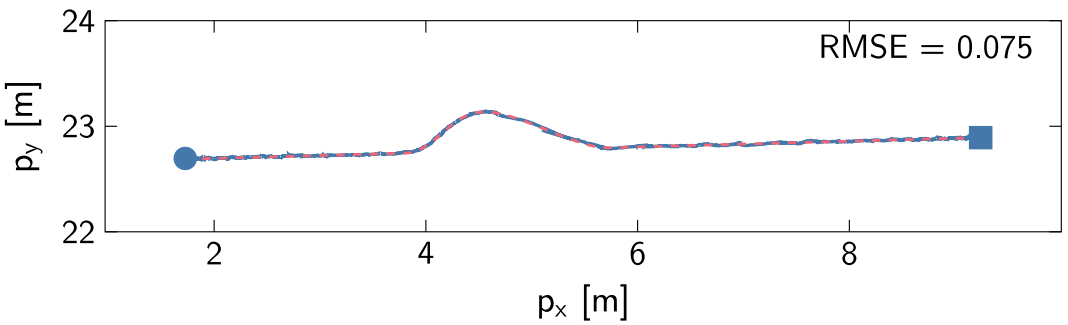
Paper



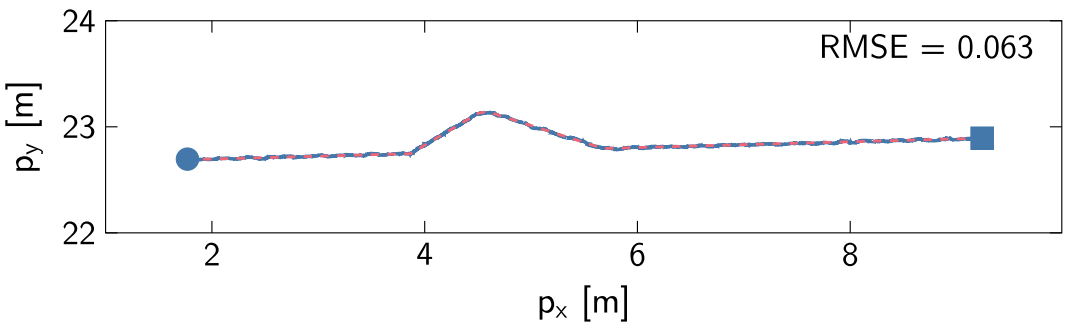
Offline planning and online tracking



(a) **Double-integrator**



(b) **Single-integrator**



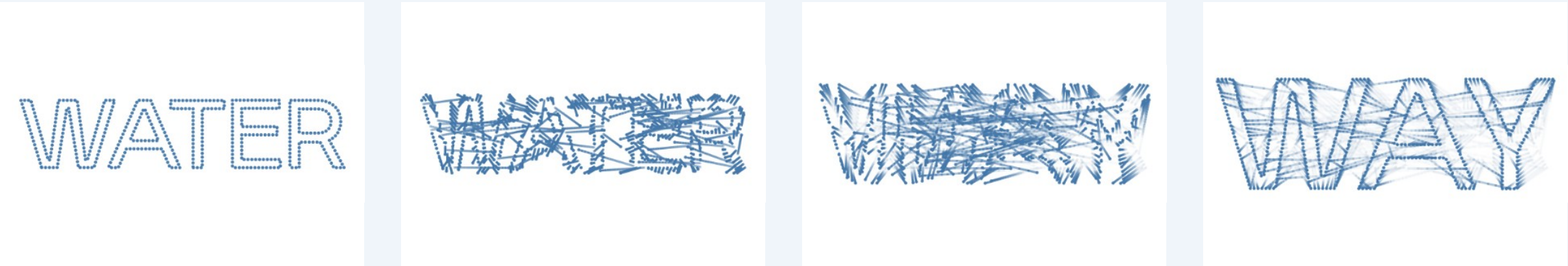
Reference Simulation Start Goal

Planning transitions for hundreds of robots

(a) **Transition snaphsots:** $of \rightarrow \star$ N = 500, K = 1000, 20 % labeled



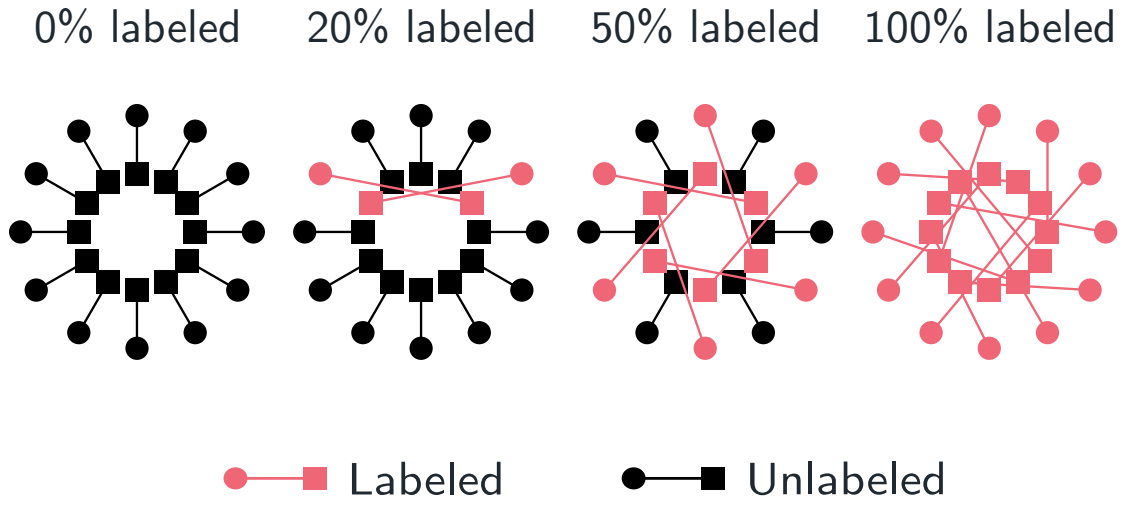
(b) **Transition snaphsots:** WATER → WAY N = 500, K = 1000, 20 % labeled



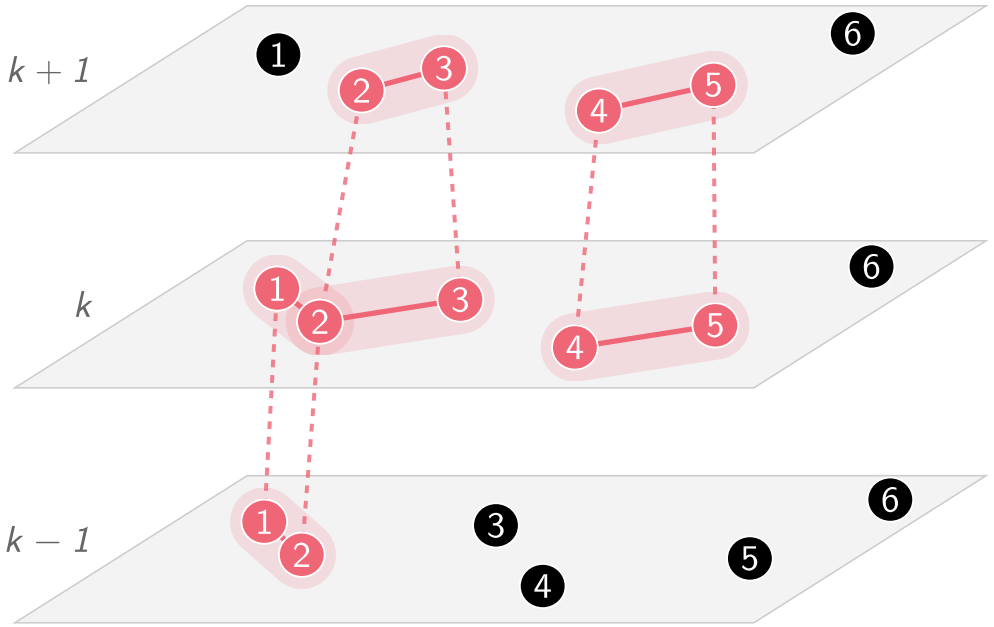
Who goes where?

Labeled specific allocation,

Unlabeled can be freely allocated (linear assignment problem)

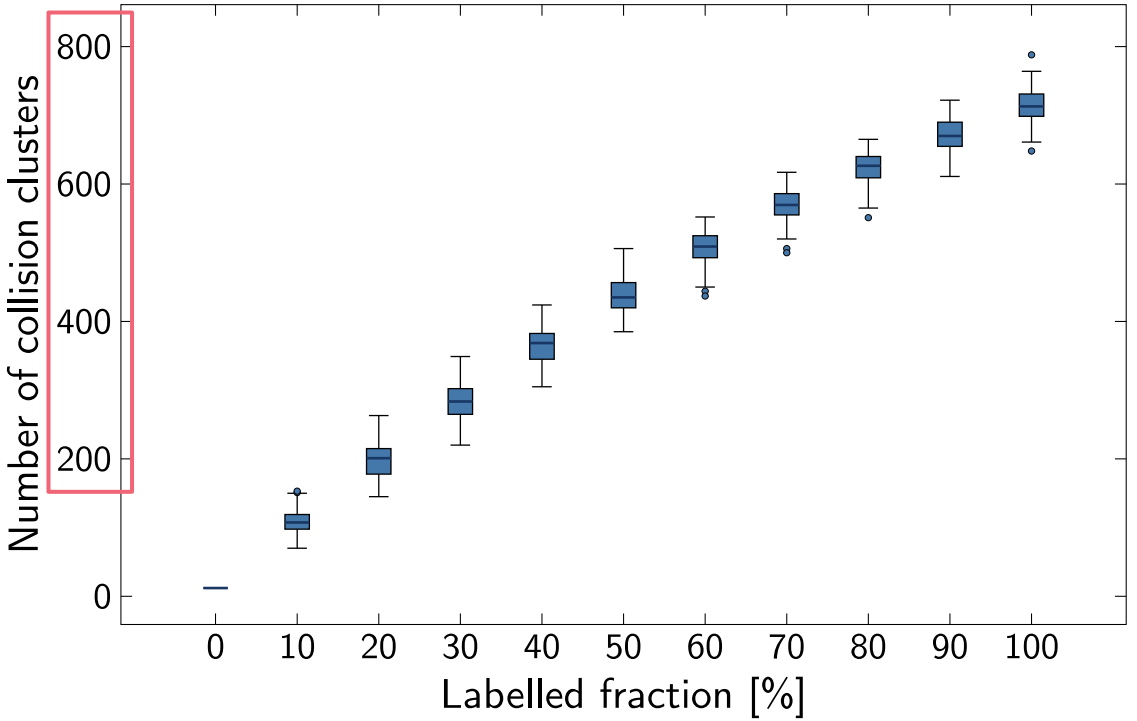


The collision graph

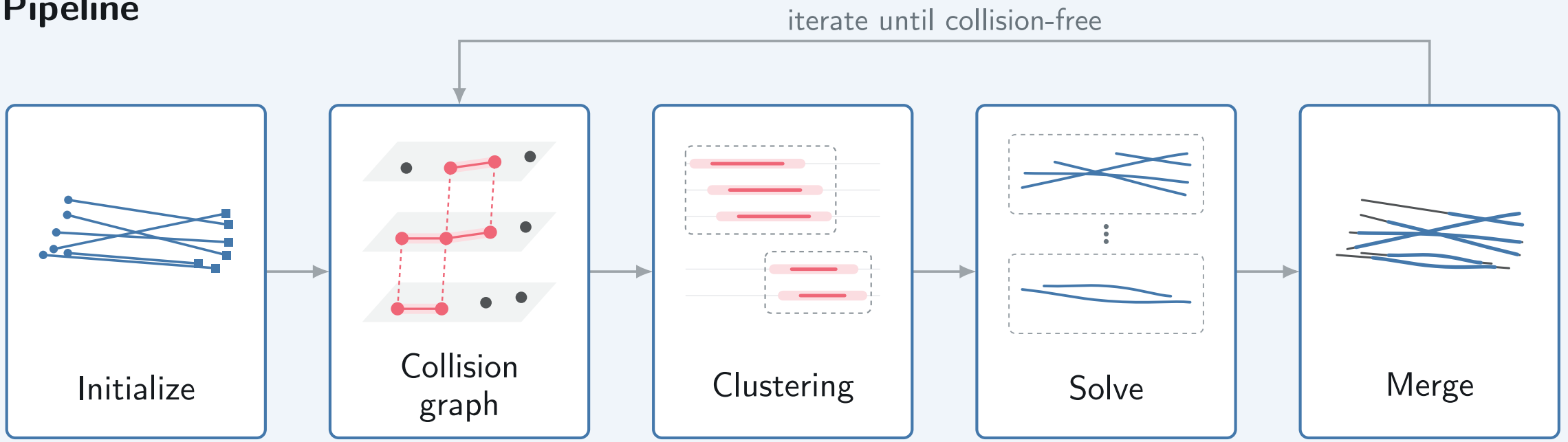


Collision clusters $\mathcal{G}_m$

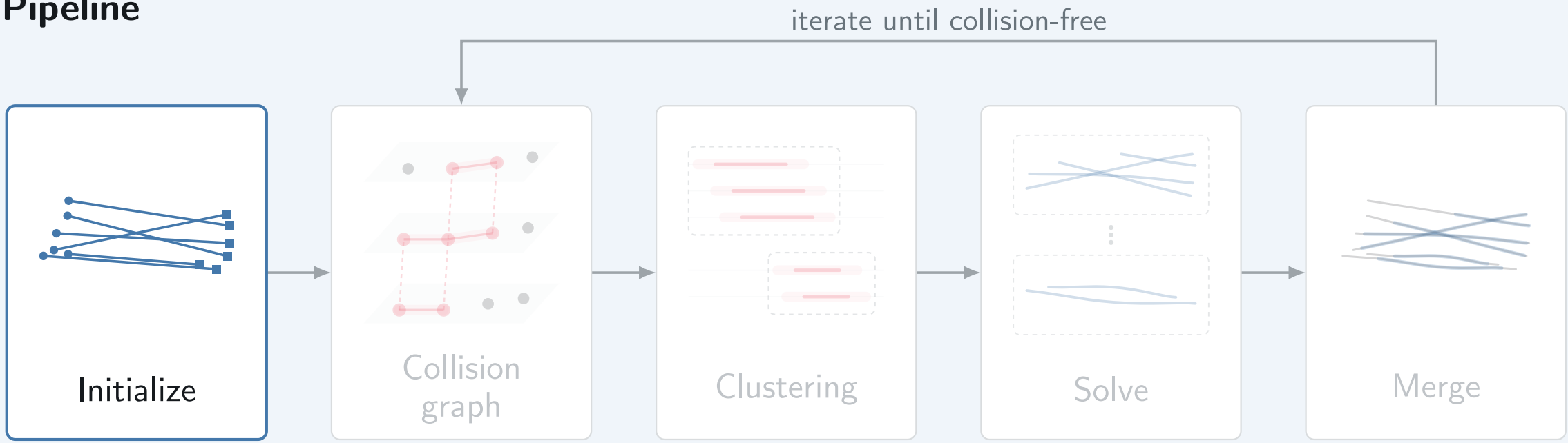
connected components of the graph



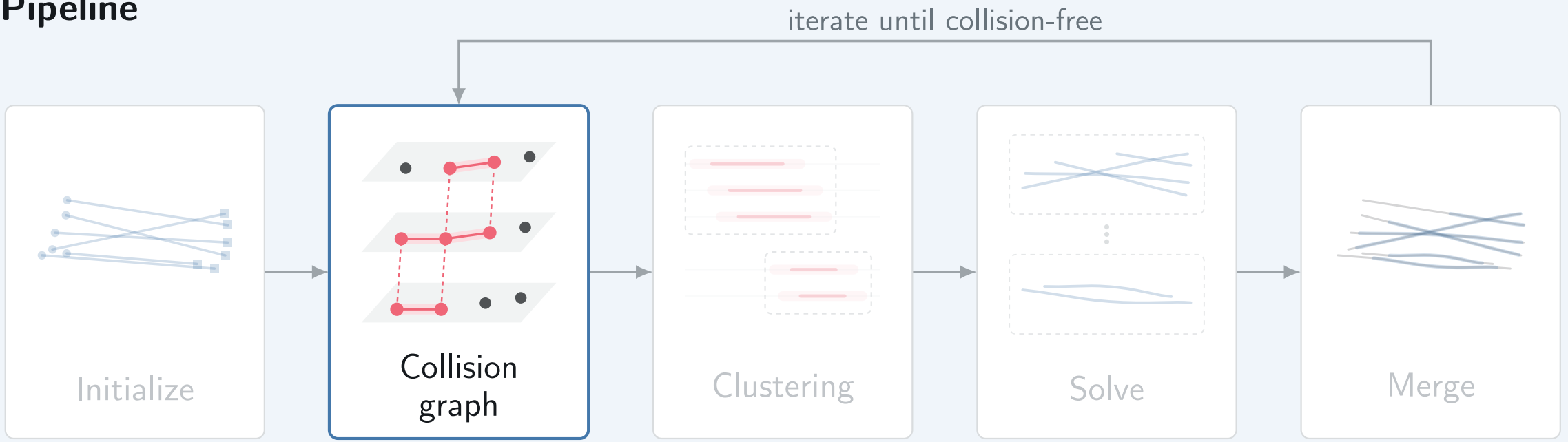
Pipeline


iterate until collision-free
Initialize
Collision graph
Clustering
Solve
Merge

Pipeline

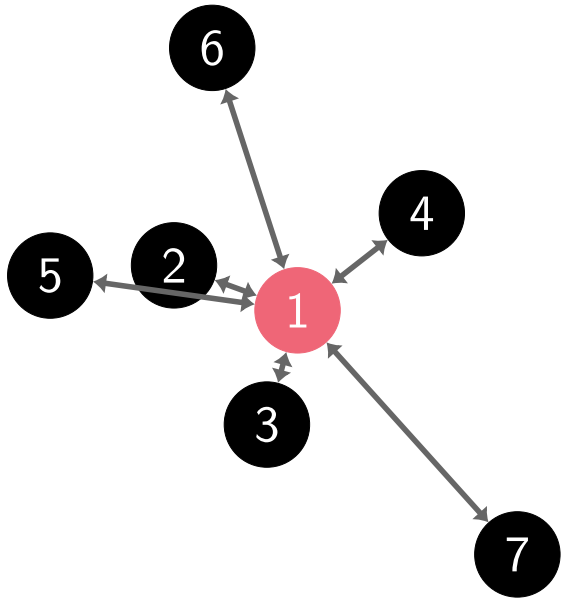

iterate until collision-free
Initialize
Collision graph
Clustering
Solve
Merge

Pipeline


iterate until collision-free
Initialize
Collision graph
Clustering
Solve
Merge

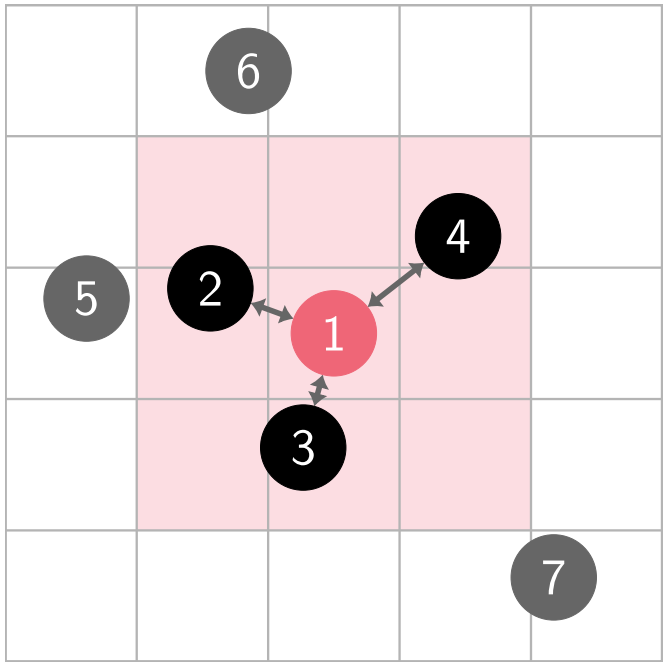
Collision detection

(a) Pairwise



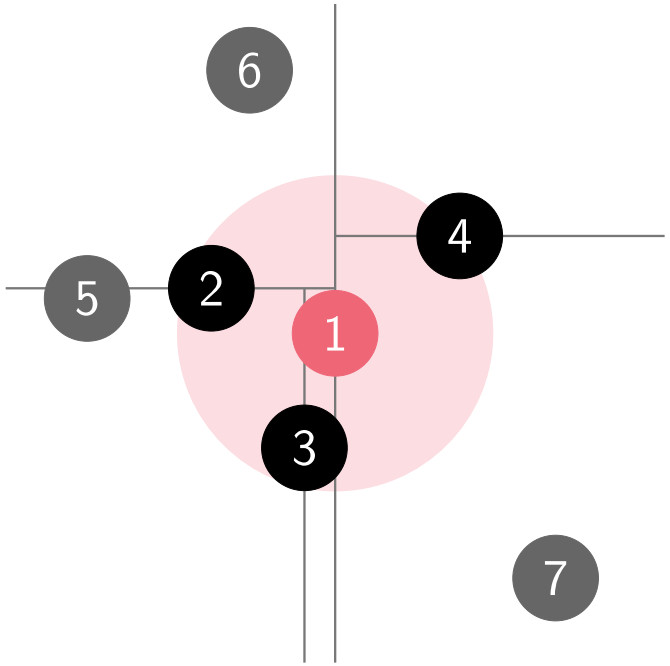
Check every pair

(b) Spatial hashing



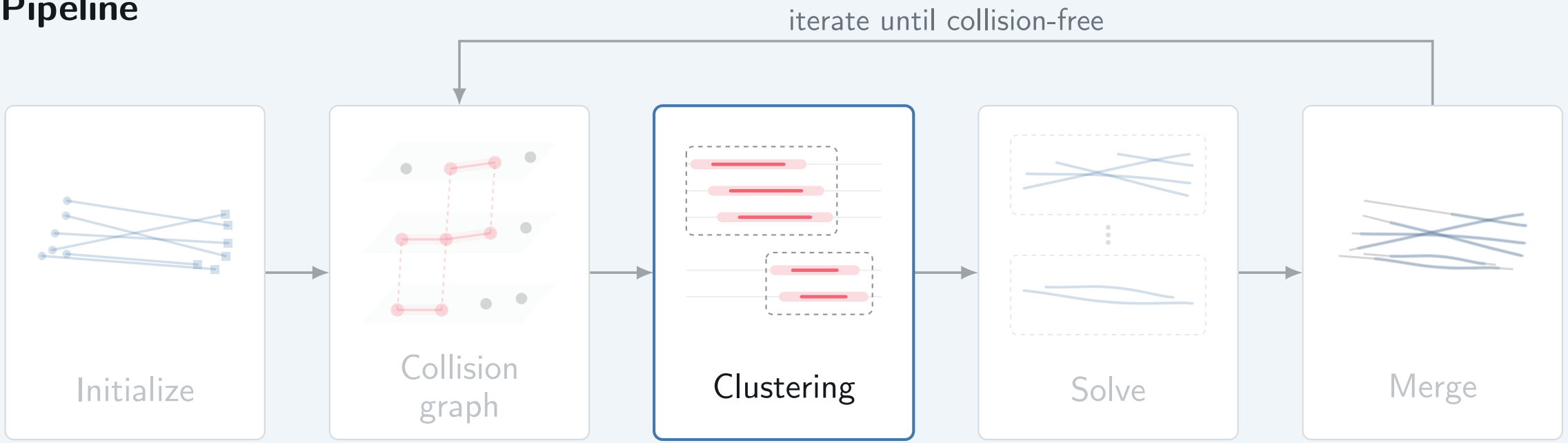
Search neighboring cells

(c) KD-tree



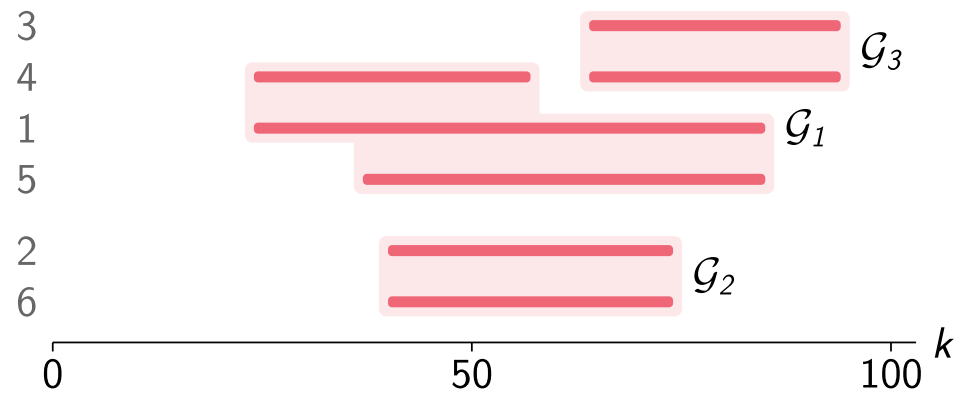
Query within a radius

Pipeline

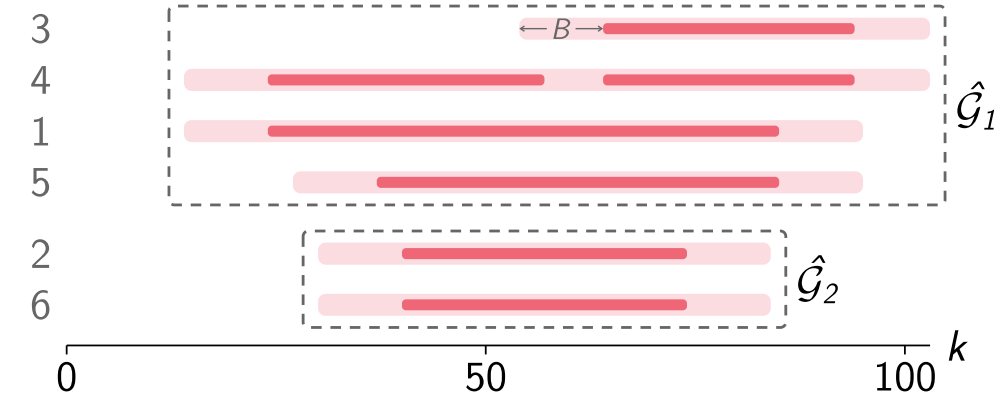

iterate until collision-free
Initialize
Collision graph
Clustering
Solve
Merge

Turning collision clusters into subproblems

Collision clusters

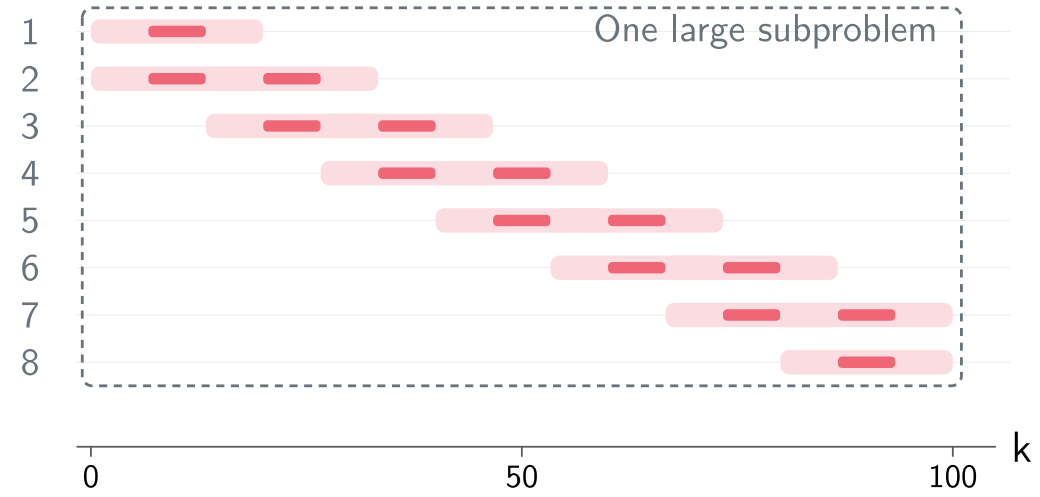


Subproblems

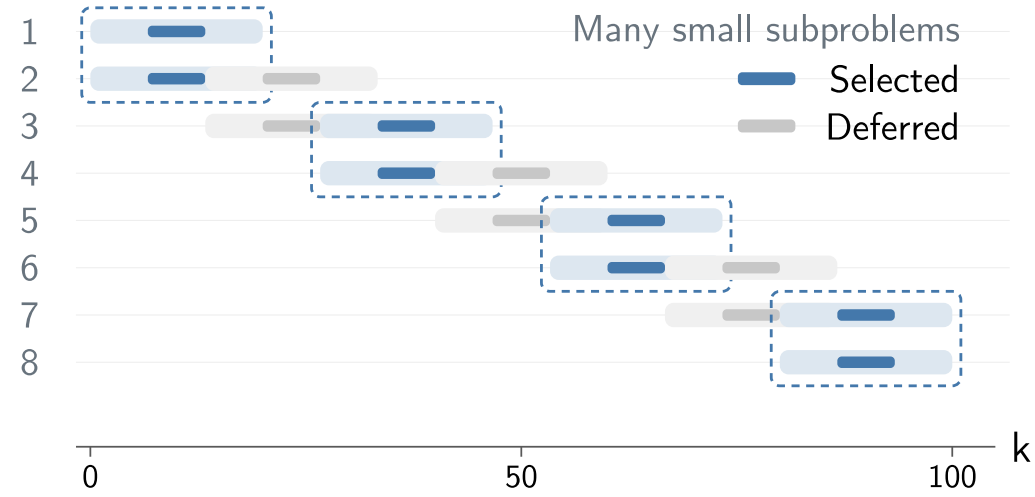


Limiting subproblem size

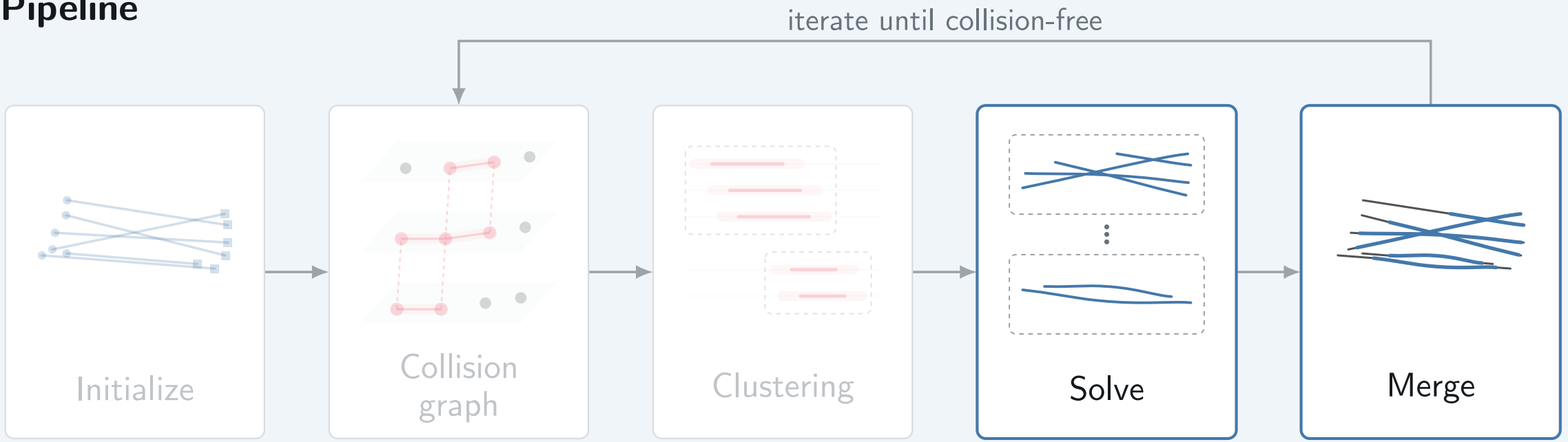
(a) Unrestricted clustering



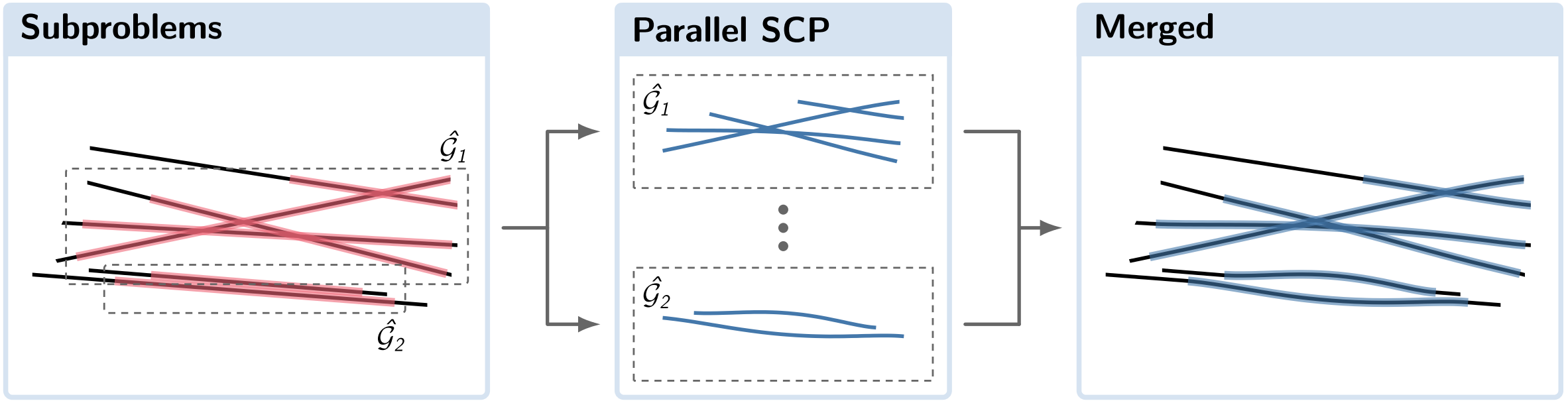
(b) Recursive splitting



Pipeline


iterate until collision-free
Initialize
Collision graph
Clustering
Solve
Merge

Subproblem solving and merging



Sequential convex programming (SCP)

Collision constraints

$$\left\| p^{(i)}[k] - p^{(j)}[k] \right\| \geq R$$

Algorithm 1 Sequential Convex Programming

1: Initialize trajectories $\bar{p}^{(i)}[k]$
2: **while** not converged **do**
3: Linearize collision constraints around $\bar{p}$
4: Solve QP $\rightarrow$ obtain $p^{(i)}[k]$
5: $\bar{p} \leftarrow p$
6: **end while**

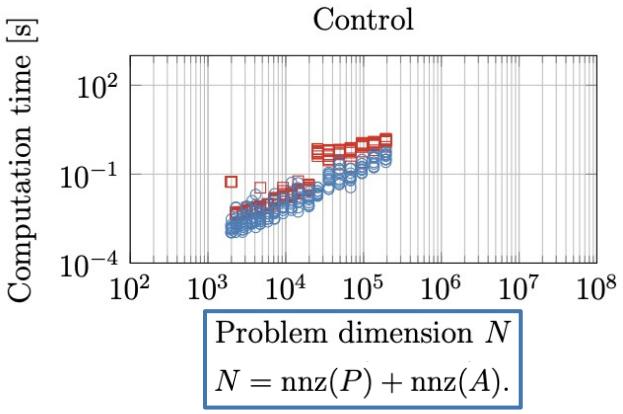
Augugliaro et al., IROS, 2012

OSQP: An Operator Splitting Solver for Quadratic Programs

Bartolomeo Stellato, Goran Banjac, Paul Goulart, Alberto Bemporad, and Stephen Boyd

February 13, 2020

$$\begin{array}{ll} \text{minimize} & (1/2)x^T P x + q^T x \\ \text{subject to} & l \leq Ax \leq u, \end{array}$$



B. Stellato et al., Math. Program. Comput., 2020.

Direct transcription

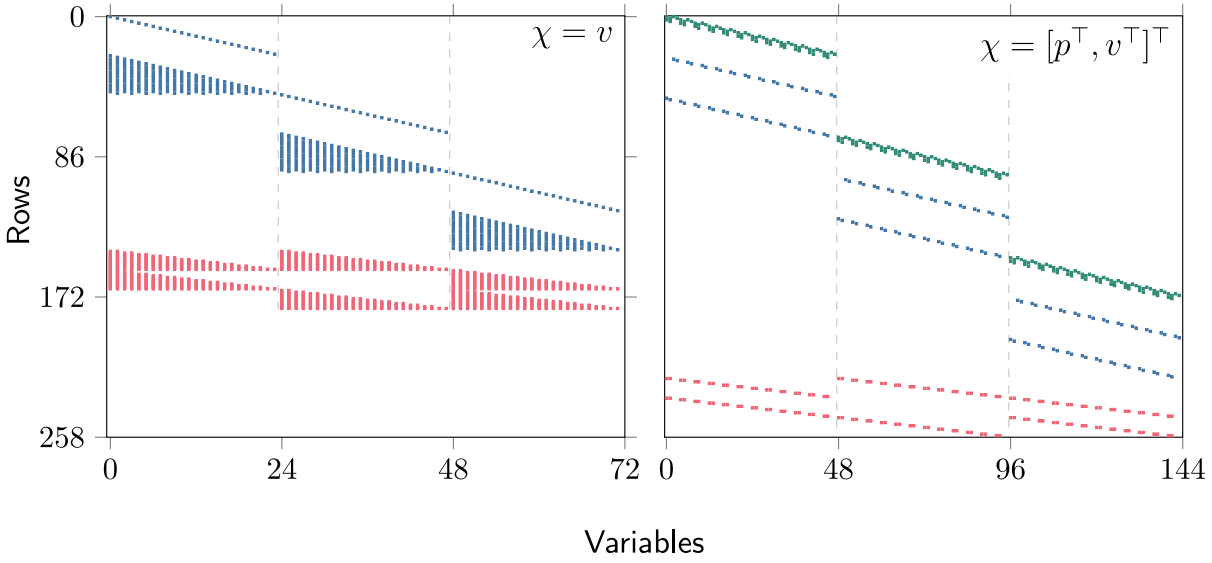
$$\min_{\chi} \|\chi\|_2^2 \quad \text{s.t.} \quad l \leq \begin{bmatrix} C_{\text{dyn}} \\ C_{\text{box}} \\ C_{\text{coll}} \end{bmatrix} \chi \leq u$$

$$\hat{n}_{ij}[k]^\top \left(p^{(i)}[k] - p^{(j)}[k]\right) \geq R$$

Linearized at previous iterate

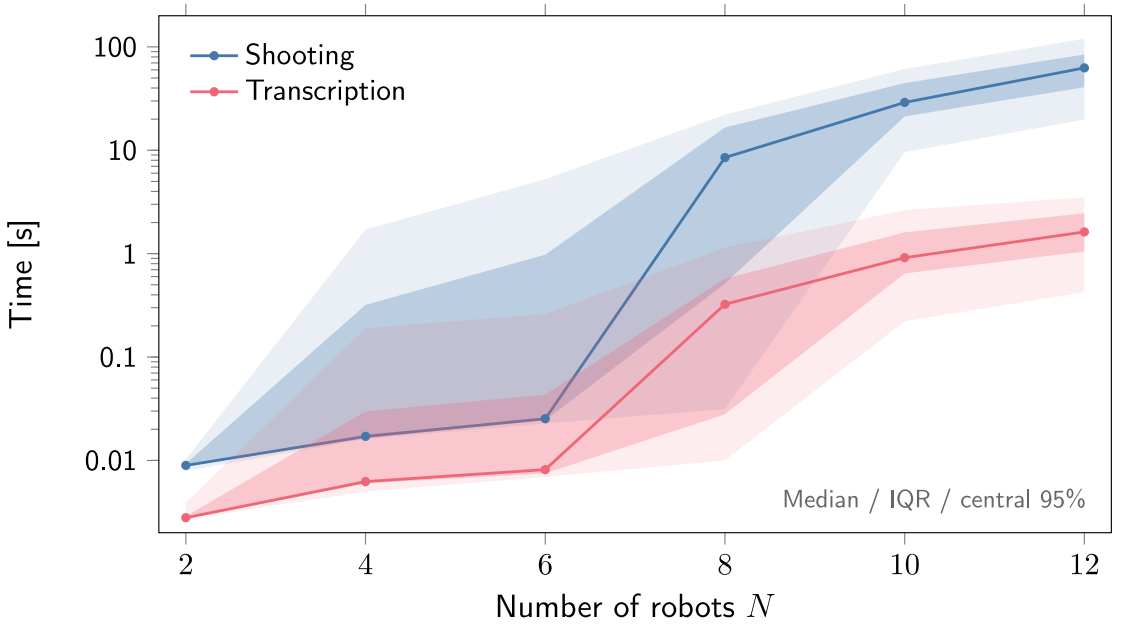
(a) Direct shooting

(b) Direct transcription



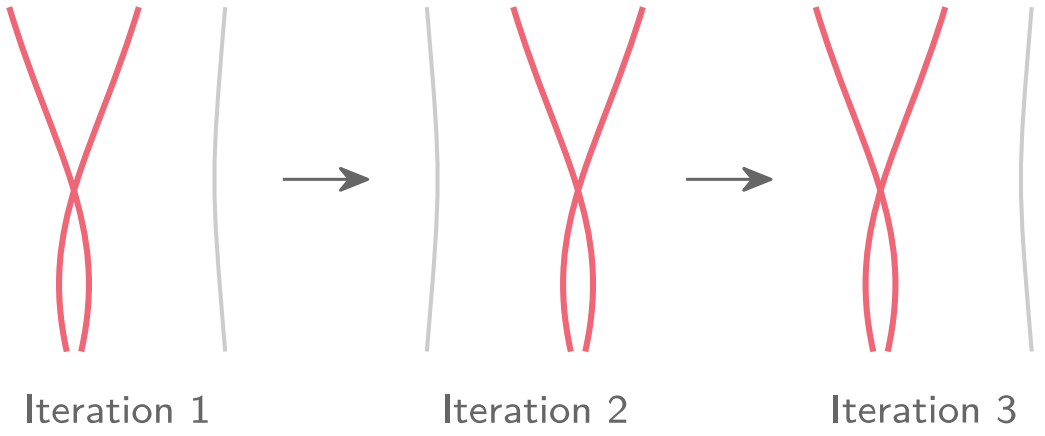
$\text{nnz}(C) \in \mathcal{O}(N^2K^2)$

$\text{nnz}(C) \in \mathcal{O}(N^2K)$

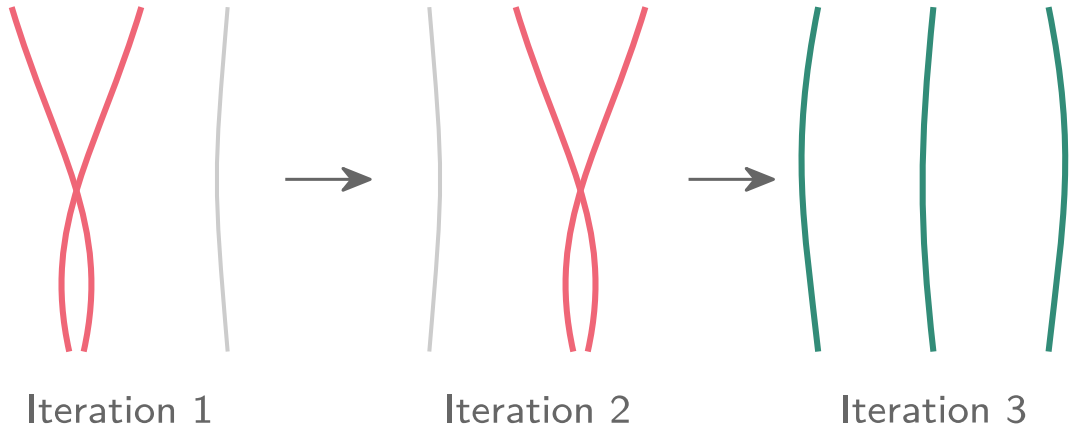


Oscillating cycles

(a) **Without cycle detection**

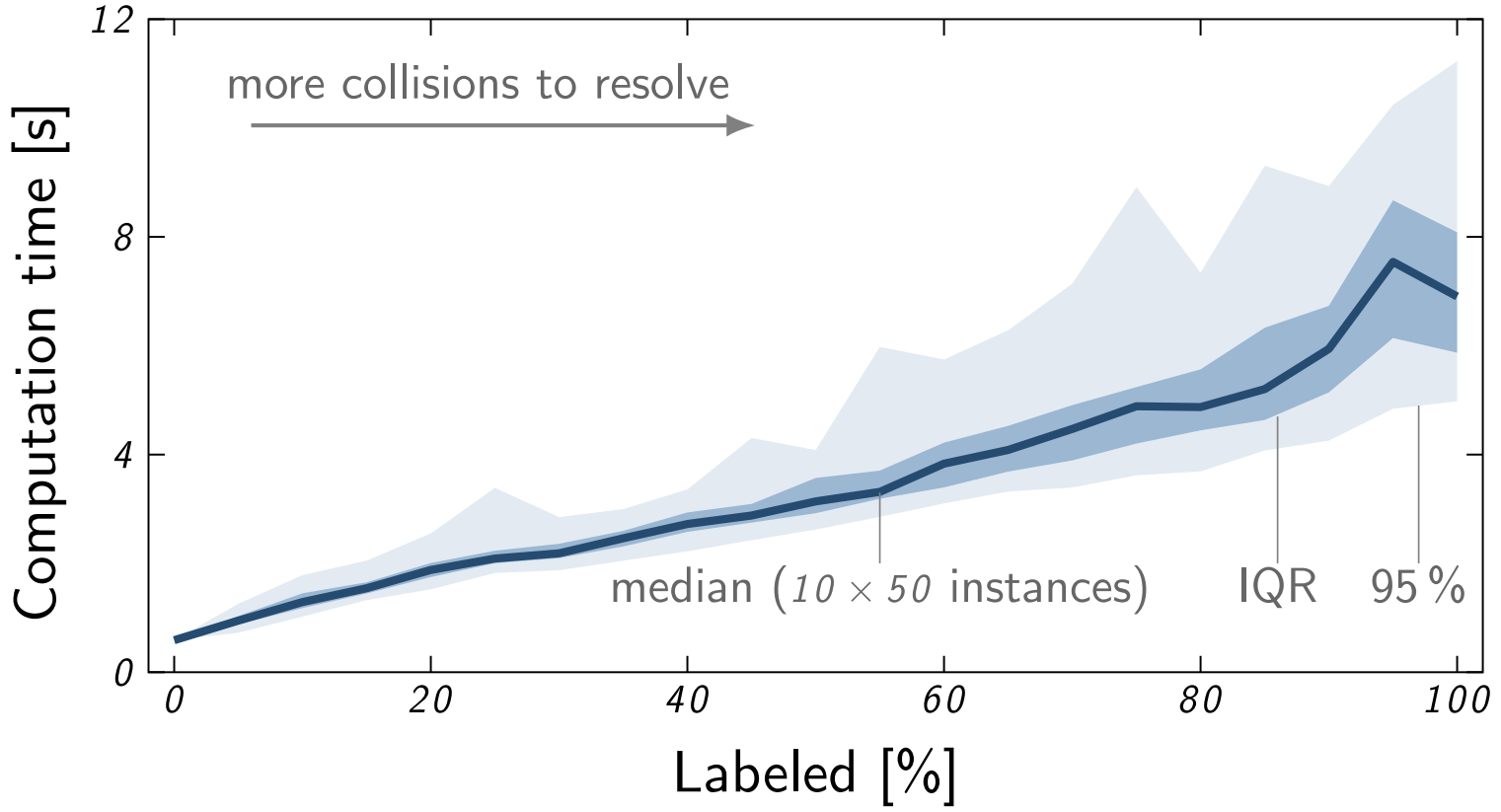


(b) **With cycle detection**



Planning for 500 robots

500 robots, 1,000 time steps



3.1 s

Median at 50% labeled

Success within 60 s

100% up to 30% labeled
99% at higher fractions

Ten transitions, 50 seeds per labeled fraction
MacBook Pro with M3 Pro, 32 GB RAM

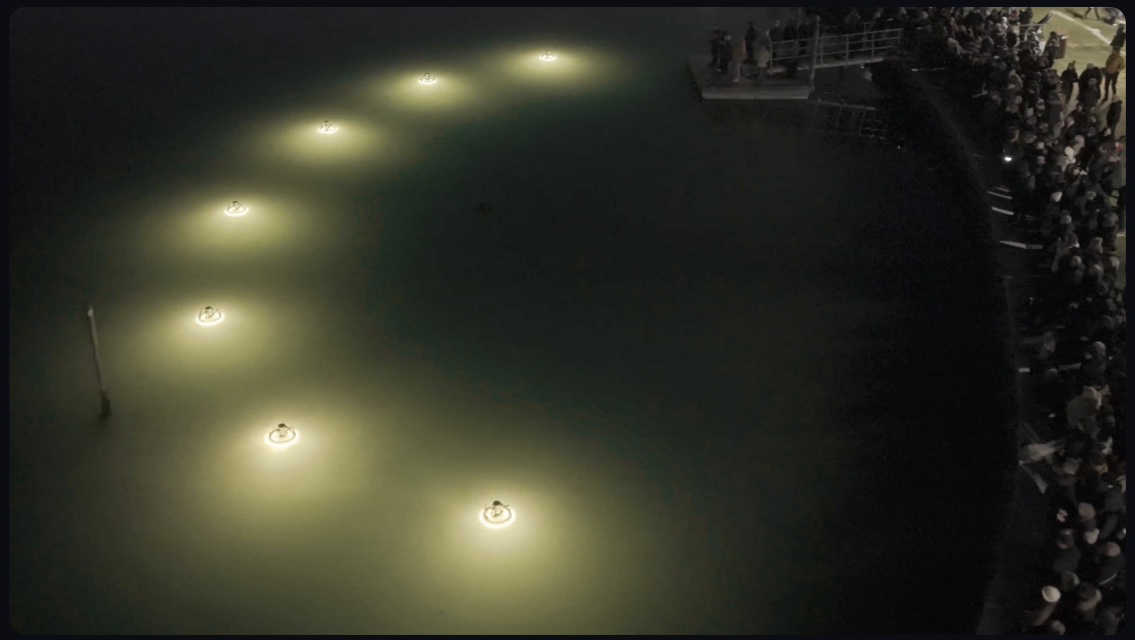
Deployment on floating robots

Lake Zürich



24-robot fleet
16 robots in planned transitions
1–2 minute transitions

Venice Biennale 2025



8 robots
2,491 time steps
Computed in 1.23 s

Thank you

Meet us at the poster to find out more.

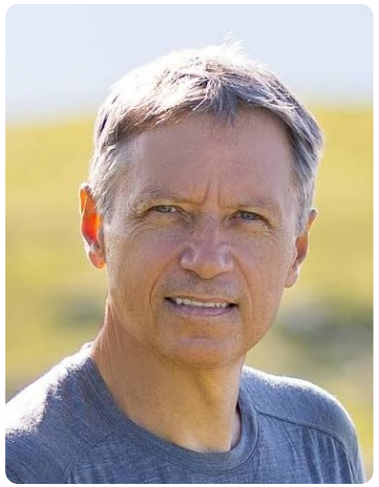
Room 324 · Pod A7 · South face



Jan
Kamm



Antonio
Terpin



Raffaello
D'Andrea



Aswin
Ramachandran

Read the paper

arXiv:2606.09620