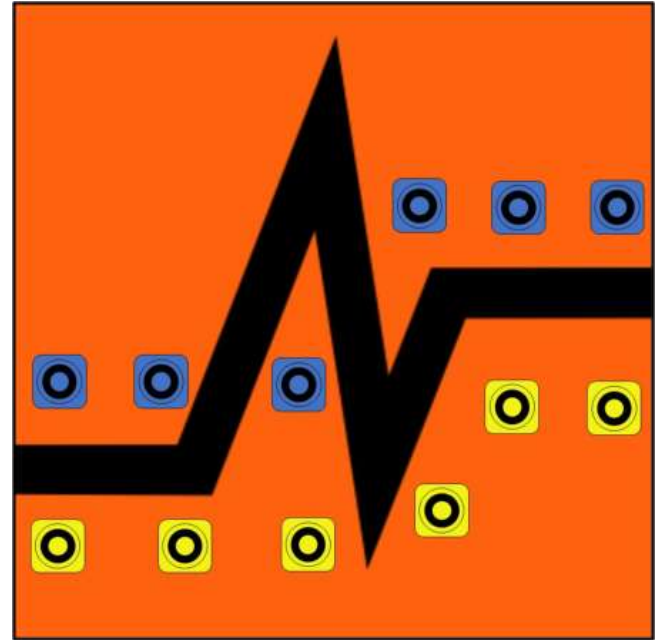


59	Dresden TU ELBFLORACE e.V.	DE	1		Best	Avg.	Last
78	Hamburg TU e-gnition Hamburg	DE	10	Finished	25.97	26.76	25.97
63	Trondheim NTNU Revolve NTNU	NO	2	DNF	29.57	29.57	29.57
31	München TU TUfast Racing Team e-Technology	DE	10	Finished	30.95	31.19	31.02
197	Schweinfurt UAS Mainfranken Racing	DE	10	Finished	41.23	41.48	41.45
41	Deggendorf IT Fast Forest	DE	2	DNF	44.16	44.16	44.16
90	Milano PT Dynamis PRC	IT	1	DNF			
54	Barcelona ETSEIB BCN eMotorsport	ES	5	DNF	29.63	33.08	29.66
277	Stuttgart DHBW DHBW Engineering Stuttgart e.V.	DE	10	Finished	36.31	39.28	36.35
44	Ulm UAS Einstein Motorsport	DE	2	DNF	74.05	74.05	74.05
172	Dortmund TU GET racing Dortmund e.V.	DE	1	DNF			
149	Erlangen U High-Voltage Motorsports e.V.	DE	1	DNF			





# How to drive your own logo at FSG



# Hellooo



## Lena Seidel

- 25 years
- Diploma student in Industrial Engineering
- Software Developer at FSD since 2023
- Member of ELBFLOACE since 2024
  - did Business Plan
  - was Head of Autonomous System 2025

if you want to connect:



You probably remember the helmet from FSG25 ^^



# What happened



2025-08-23

08:30 am: FSG changed a track after receiving

10:38 am: Started Trackdrive

10:40 am: AS Alumni chat

Niklas Leukroth 10:40 Uhr  
alex ich wills nicht hören!!!

Alexander Phieler 10:41 Uhr  
ICH HABS GESAGT

1

10:43 am: Finished Trackdrive

10:44 am: First Meme generated

Car Team	Last	Best	A	A	Lap
59 DE Dresden ...	29.65	29.63	0	10	10

Lukas 10:44 Uhr  
Sind das diese 10 for 10s?

1000045189.jpg



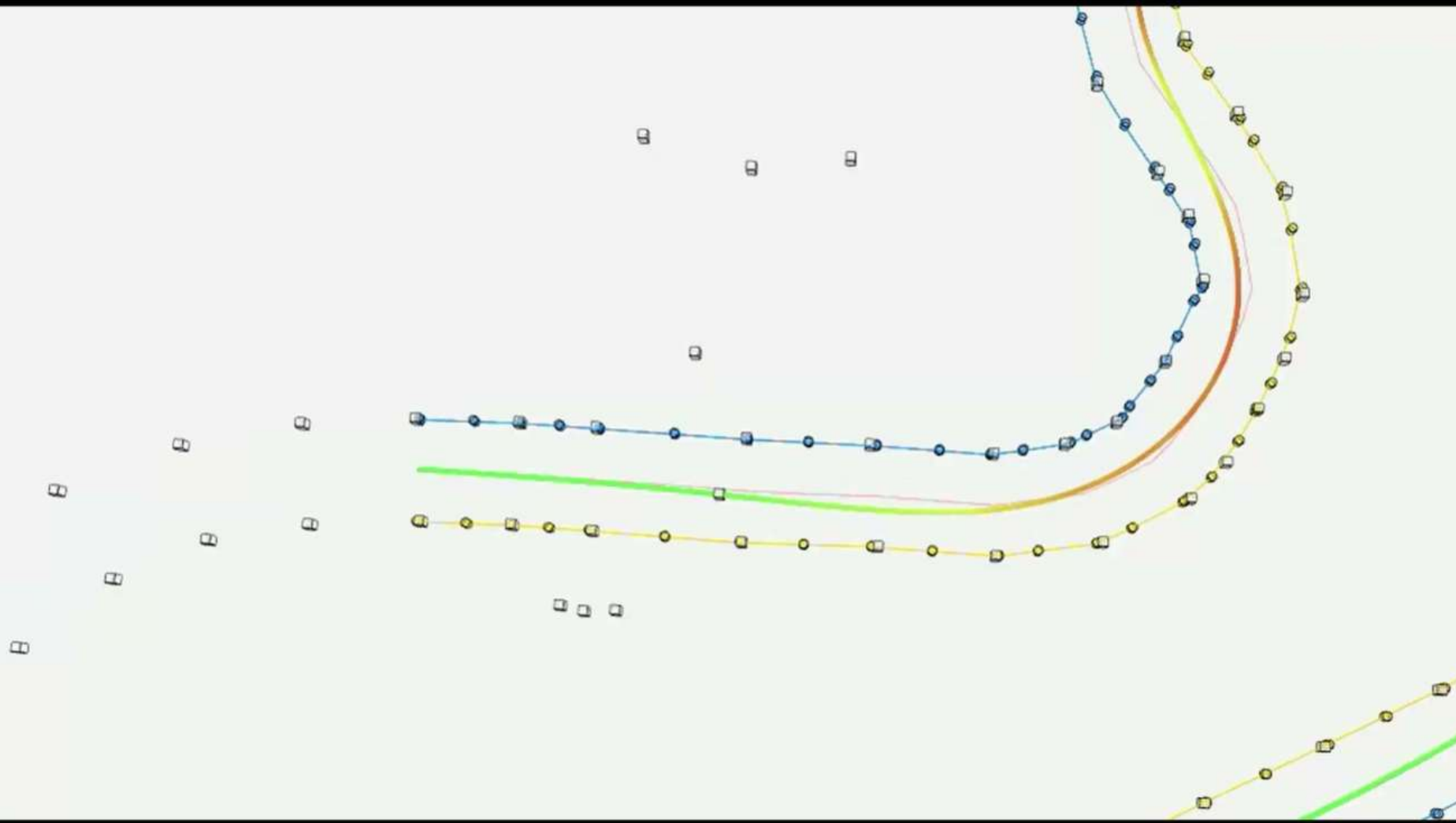
17

Jim 10:46 Uhr  
File



28 39

2 Antworten Letzte Antwort vor 7 Monaten



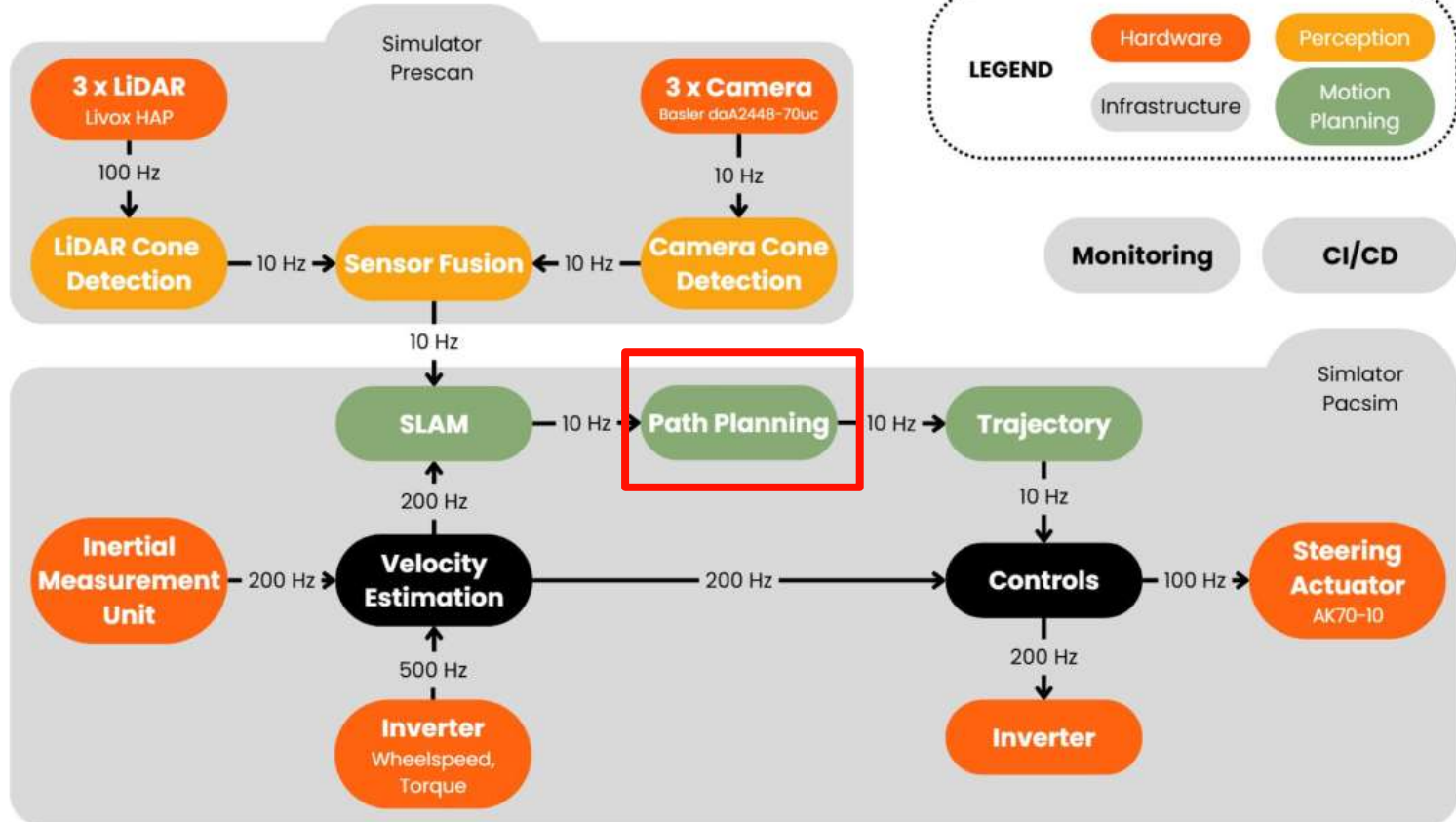
# Our Perception Setup



FSG 2025: 3 Livox LiDAR HAP TX → 300° hFoV

- LiDAR only Path Planning
- no color information

# System Overview





**Input:** Map of cones from SLAM, pose

**Output:** Centerline with normals and width

- **Long paths (>30m)**
- Runtime < 100ms
- Hardware: LiDAR only perception concept (last season)
  - needs to work without color information
  - robust against false positives



## Concept Comparison

Delaunay triangulation based	Search based lane segmentation	SVM	End-To-End
<ul style="list-style-type: none"><li>• Efficiency</li><li>• Works without colors</li></ul>	<ul style="list-style-type: none"><li>• Works without colors</li><li>• Robust against false positives</li></ul>	<ul style="list-style-type: none"><li>• Efficiency</li><li>• Robust against false positives</li><li>• Robust against false negatives</li></ul>	<ul style="list-style-type: none"><li>• Potential efficiency</li><li>• Potential performance</li></ul>
<ul style="list-style-type: none"><li>• Sensitive against false positives</li><li>• Sensitive against false negatives</li></ul>	<ul style="list-style-type: none"><li>• Efficiency</li><li>• Sensitive against false negatives</li></ul>	<ul style="list-style-type: none"><li>• Doesn't work without color information</li><li>• Sensitive to wrong colors</li></ul>	<ul style="list-style-type: none"><li>• How to formulate?</li><li>• How to ensure generalization?</li></ul>

# Path Planning Concept



## 1. Starting Cone Search (SCS)

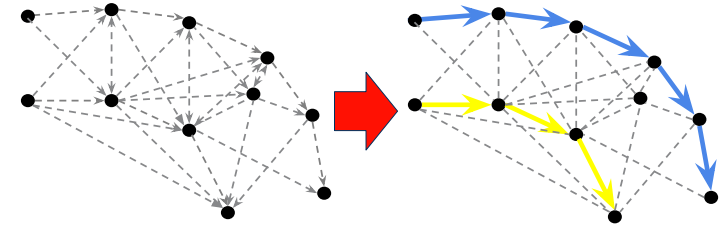
build an initial left/right lane seed

## 2. Lane Detection / Lane Rating

## 3. Centerline Generation

Delaunay triangulation + midpoint ordering

## 4. Global Path Connection



Augsburg paper: [2405.16369](#)

# Starting Cone Search



Idea: Generate candidate start lanes from the currently visible landmarks, each candidate consists of **6 cones**:

- Build candidate lanes from **3 left + 3 right cones**
- Remove implausible candidates using basic geometric checks
- Rate the remaining candidates with a cost function
- Select the **lowest-cost lane** as the search seed

# Why SCS is needed ?



- during normal operation, the **existing lane / tail** gives the search direction
- At the **beginning of the track**, this tail does not exist yet
- Therefore we need a local initialization step that answers:
  - which cones are likely left border?
  - which cones are likely right border?
  - what is the initial lane direction?



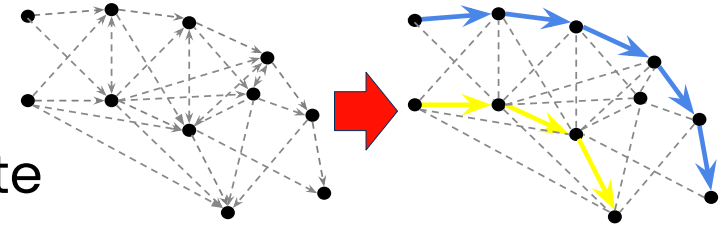
## the preferred lane should:

- start **close to the car** (20 m)
- have left and right boundaries with **similar heading**
- point roughly **forward**
- consecutive cones on one side must be within **5.3 m**
- have a mean track width close to **3 m**
- form roughly **rectangular lane geometry** at start and end

$$\text{totalCost} = 3.0 \cdot \text{startPosCost} + (\text{angleDiff})^2 + 0.2 \cdot (\text{relativeDiff})^2 + 10.0 \cdot \text{distanceVariance} + |\text{meanWidth} - 3| + 0.2 \cdot (\text{startCornerAngle} + \text{endCornerAngle})$$

# Lane Detection / Rating

- Use the starting lane as the initial state  
search direction is based on **tail extrapolation**
- Expand left and right borders step by step
- Use graph-based search with:
  - hard constraints
  - cost-based rating
  - pruning / branch-and-bound



$$cost = \frac{\sum w_i \times term_i}{length} - 10.0 \times length$$

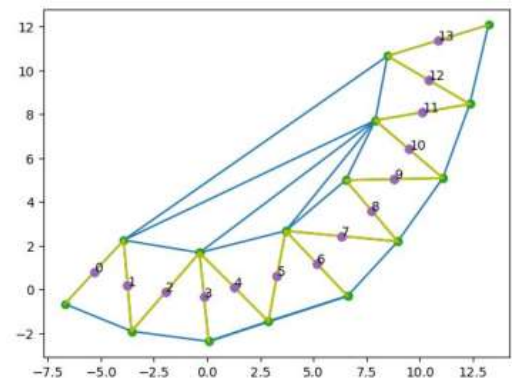
**Goal:** find the **best feasible local lane** within limited runtime

# Centerline Generation



The centerline is generated by:

- Delaunay triangulation over left and right lane
- keep only cross-border edges below 8 m
- compute edge midpoints
- order **midpoints** by nearest-neighbor recursion
- possibly reverse or reorder the beginning of the path





## Connection to the Global Path:

- The main problem appears **after the first lap**
- The challenge is the **loop detection**
- The planner must decide:

when the current lane should be treated as a global path  
whether start and end are allowed to connect  
how the resulting midpoints are ordered



## Connection to the Global Path:

- connection logic works on the current ordered lane
  - these are the lane borders that already came out of the lane search → every cone gets a lane index
- considers cones that are currently at the beginning or end of the ordered lane



- Trackdrive started in a corner
  - we assumed a straight line at the start
  - so we changed the tangent/angle-based trigger of end and start path
  - to majority matching check: **3 out of 5 cones are enough**
- it was implemented at FSEast at night with a beer 🍺

Aug 02, 2024



Implement majority global path connection

Lunas21 authored Aug 2, 2024

Aug 01, 2024



actual good GlobalPath check for overlap

Lunas21 authored Aug 1, 2024



overhaul globalpath trigger

Lunas21 authored Aug 1, 2024



For a possible global-path closure, the centerline stage may connect cones that are:

- near the **start/end region of the current ordered lane**

More concretely:

- **3 consecutive cones** of a border need to match **first 3/5** cones of the same border

→ BUT, this is still not the problem of our FSG25 TD



## Why not using the Starting Cones ?

- 3+3 (6) cones leave only a few options
- cones can be missed by perception or SLAM  
order of cones can be slightly different
- **it's more robust to consider the whole and current lane and define the start and end of it**



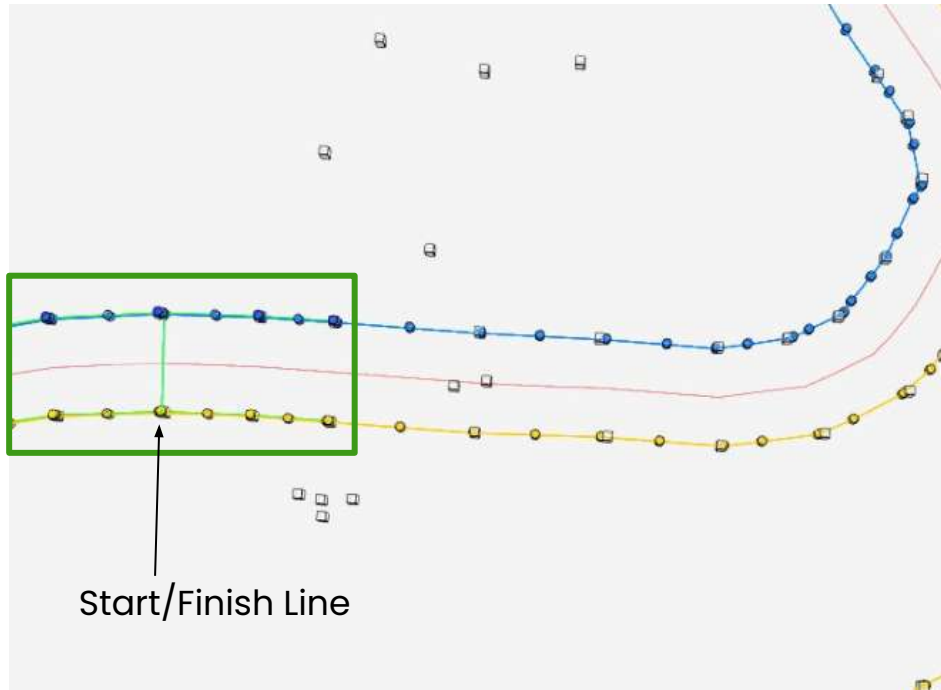
## Failure can happen because:

- start/end tangent angle fails the potentialGlobalPath test ( $< 60^\circ$ )
- first/last cones are not matched robustly enough
- special start/end cone zones do not align with actual lane ordering
- midpoint ordering near the start is unstable
- visited handling and sequence matching are heuristic
- small changes in lane length or cone order shift the connection logic
- start is a curve ?

# Global Path Connection



Here you can see the perfect matching of the 3/5 cone set after the start/finish line → so loop closure worked fine

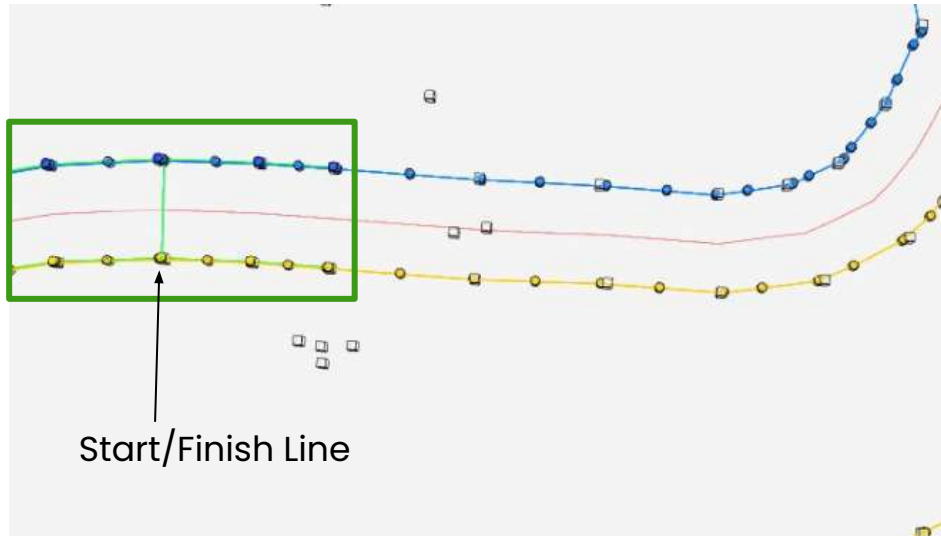


Where is the problem ?

# THE PROBLEM



BUT our path planning is still running while computing that we found the Global Path



Path Planner:

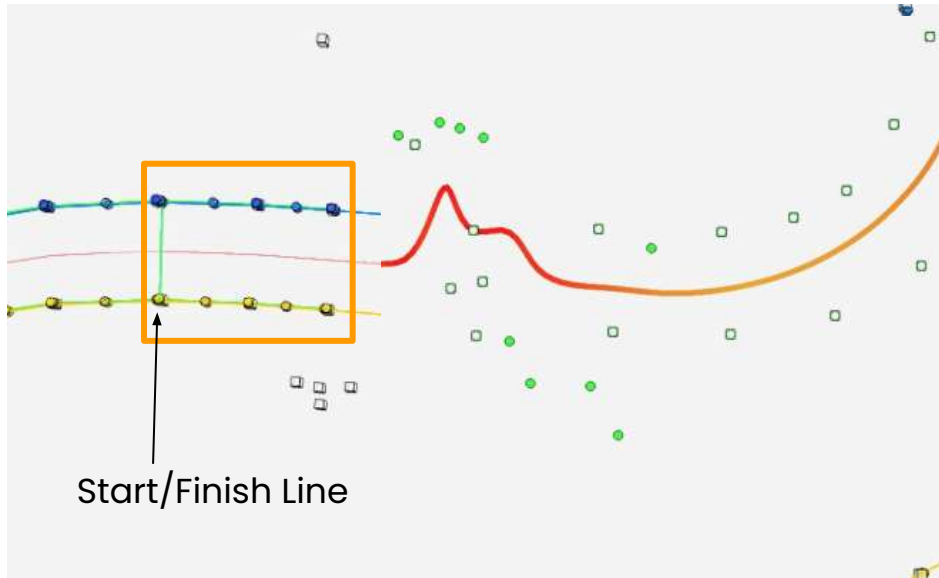
**I'M FAST AS FUCK**



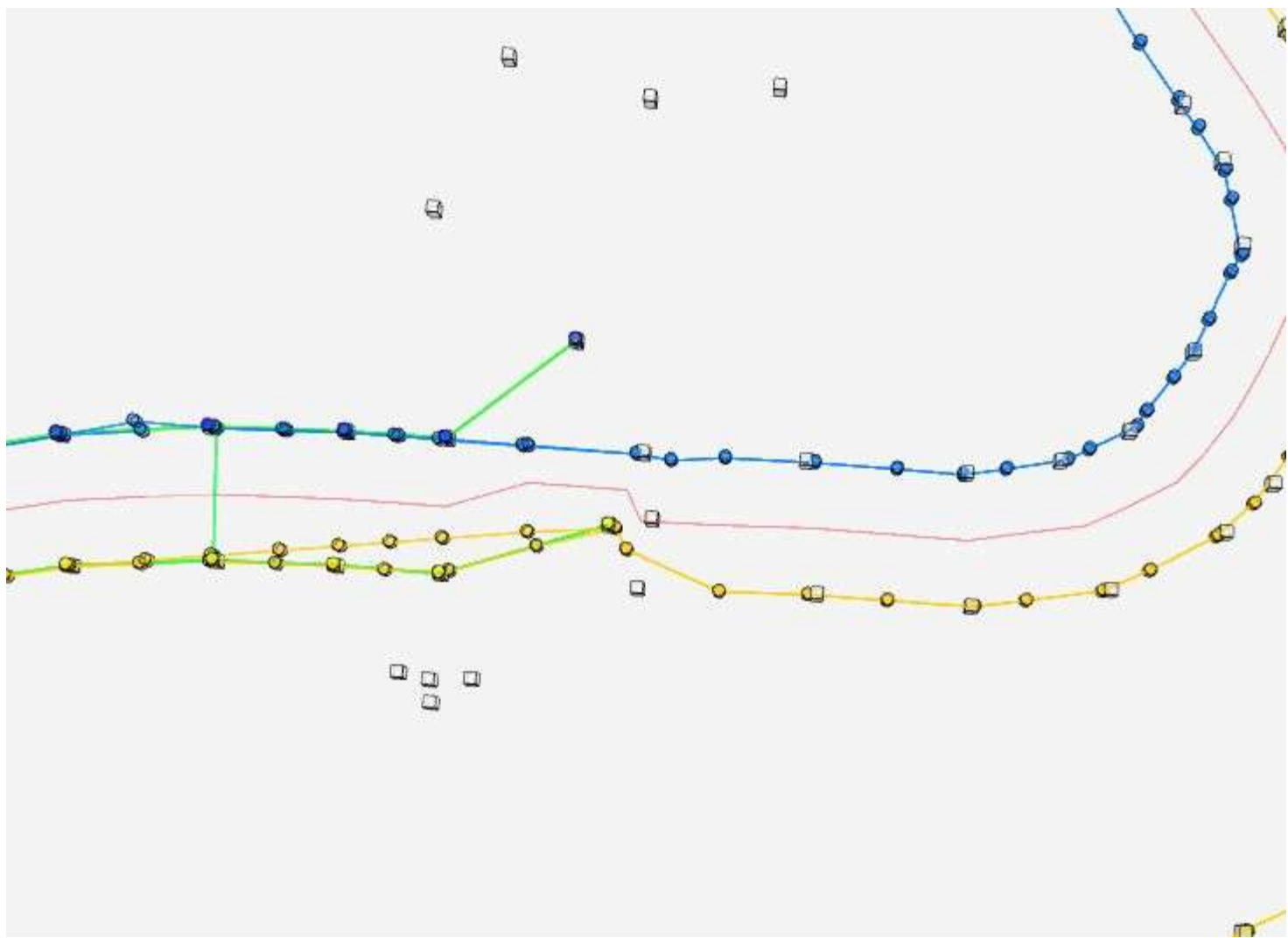
# THE PROBLEM

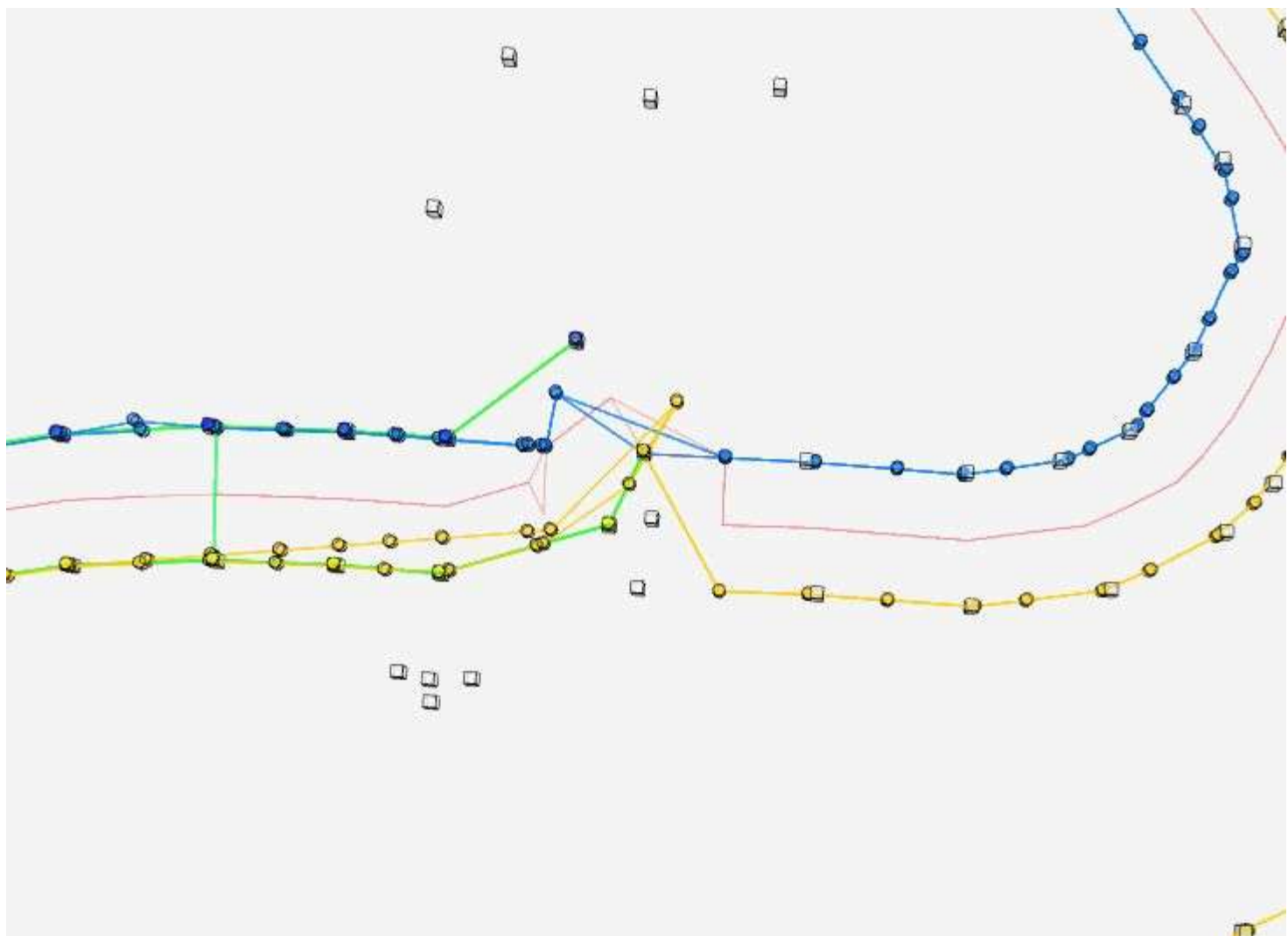


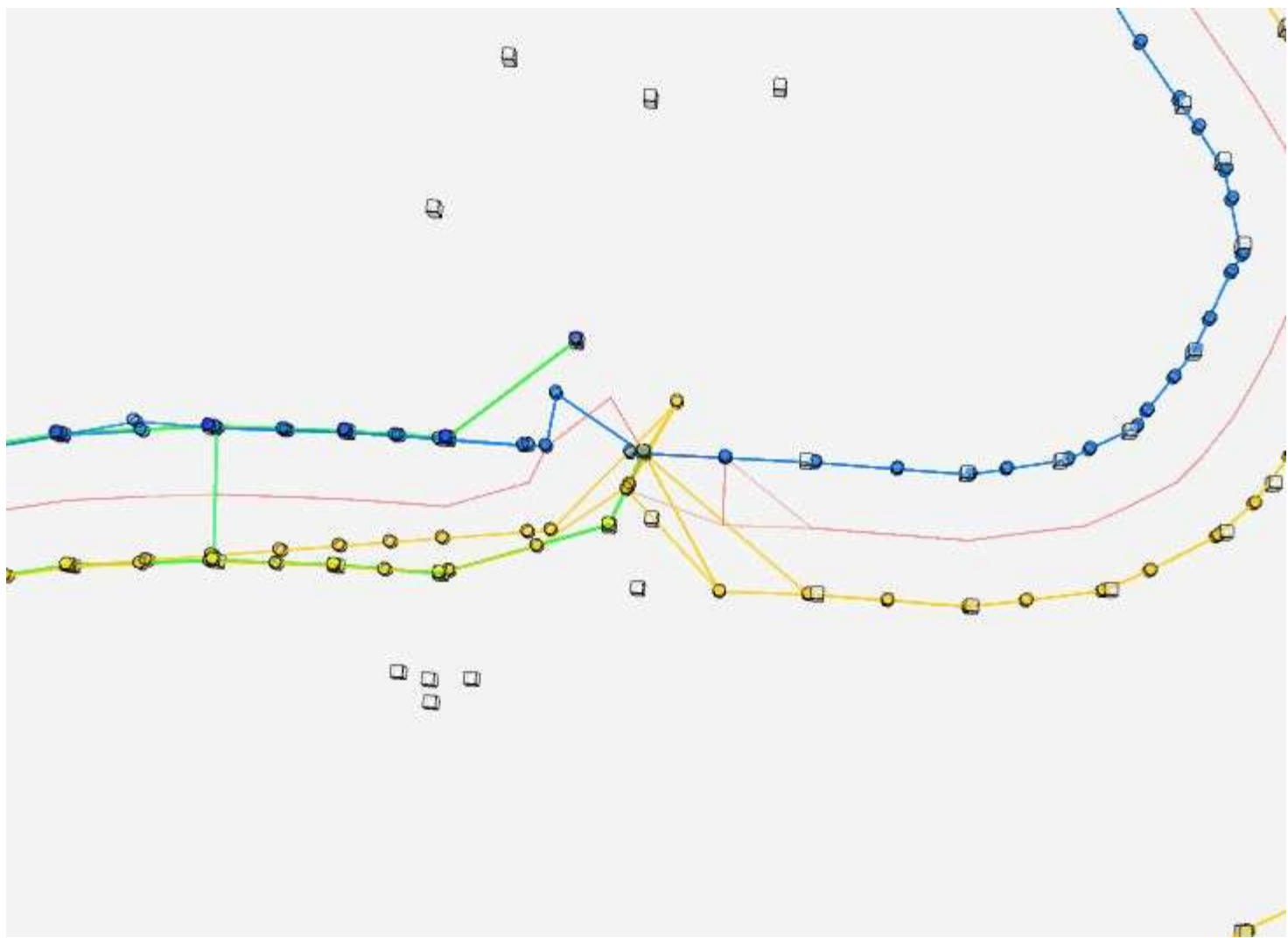
you cannot remap a cone already included in the path, except the first 5 cones after the start/finish line, so path planner needed to work with leftover SLAM landmarks

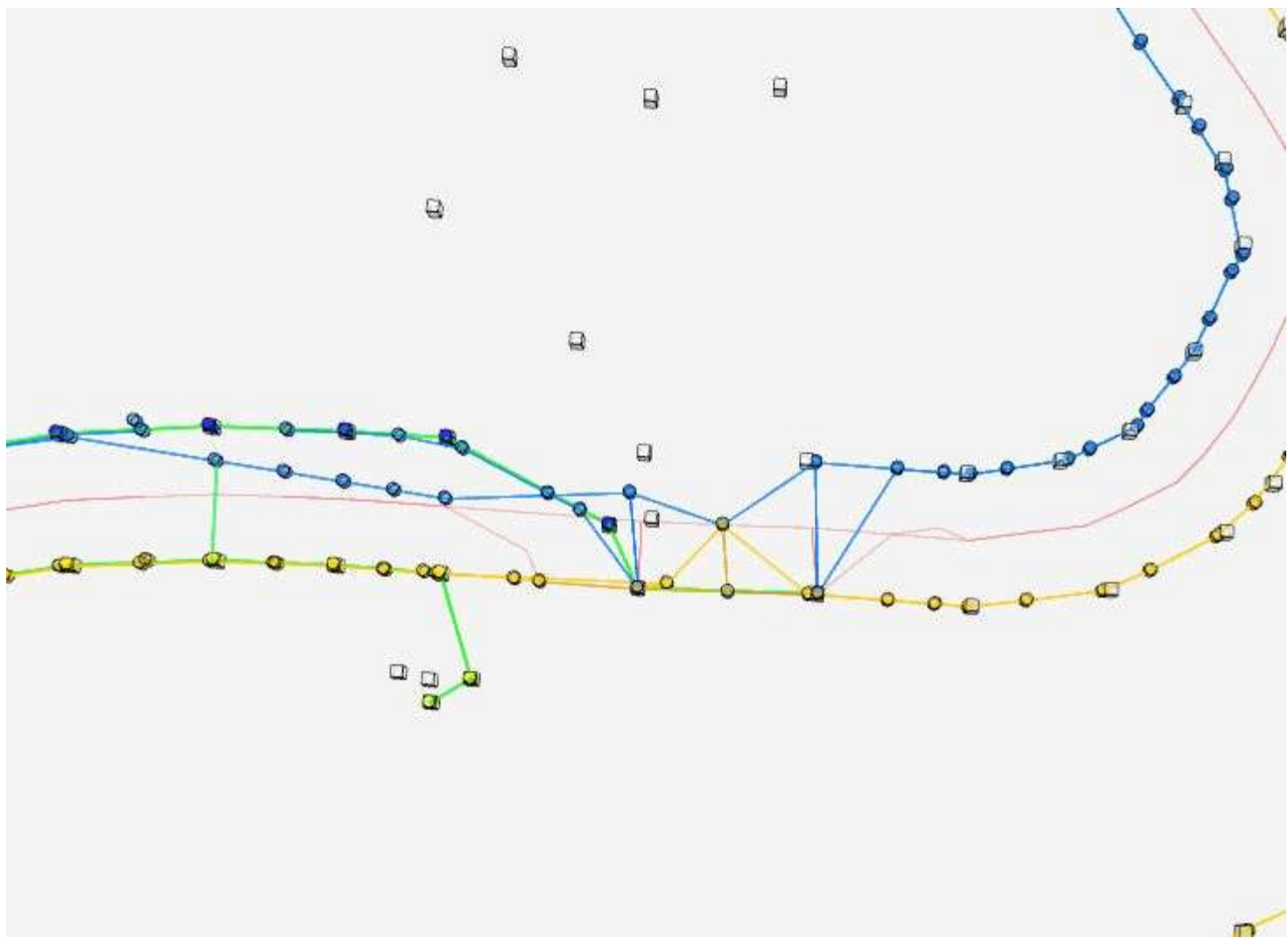


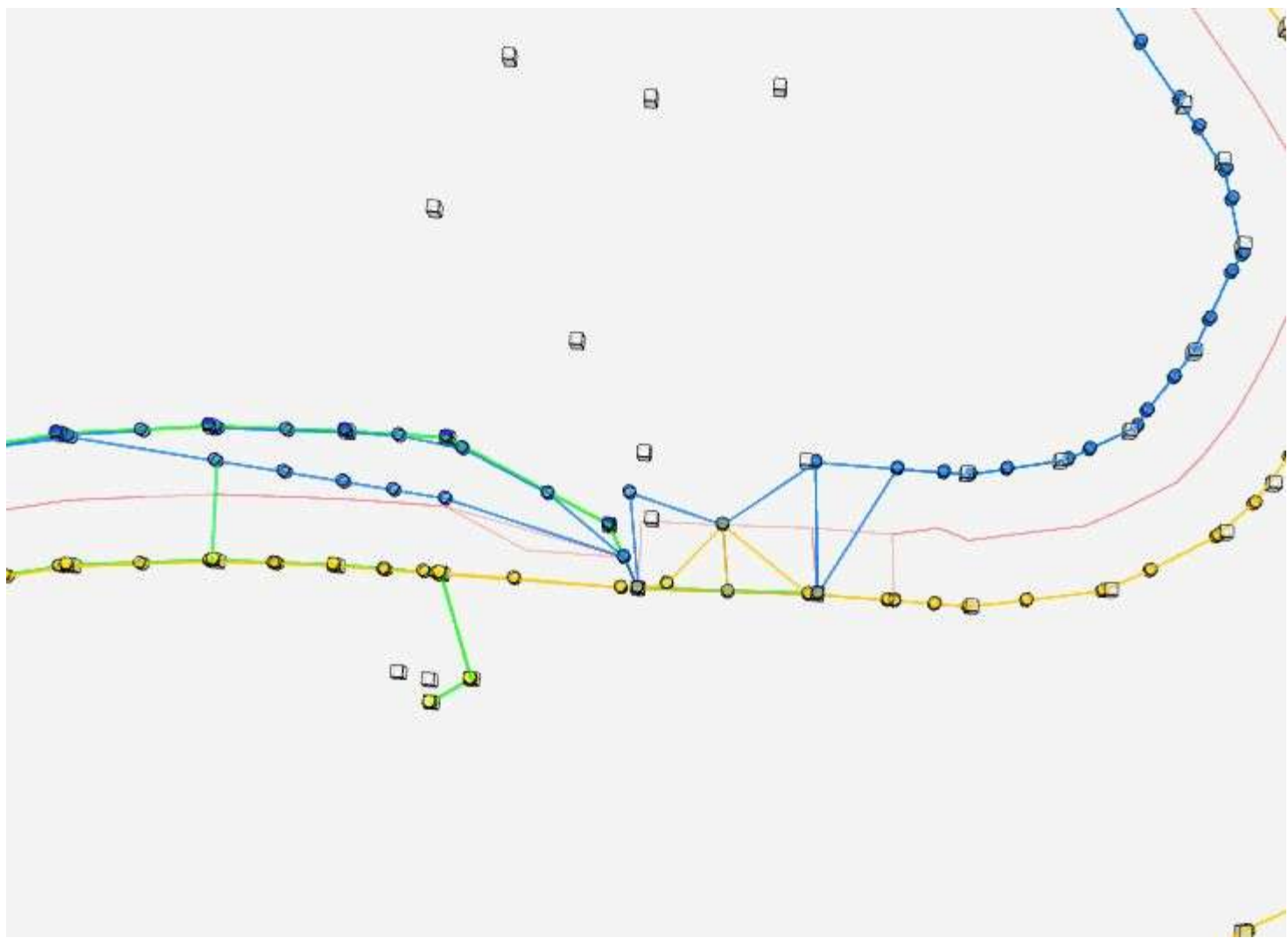
what am i supposed to do ???

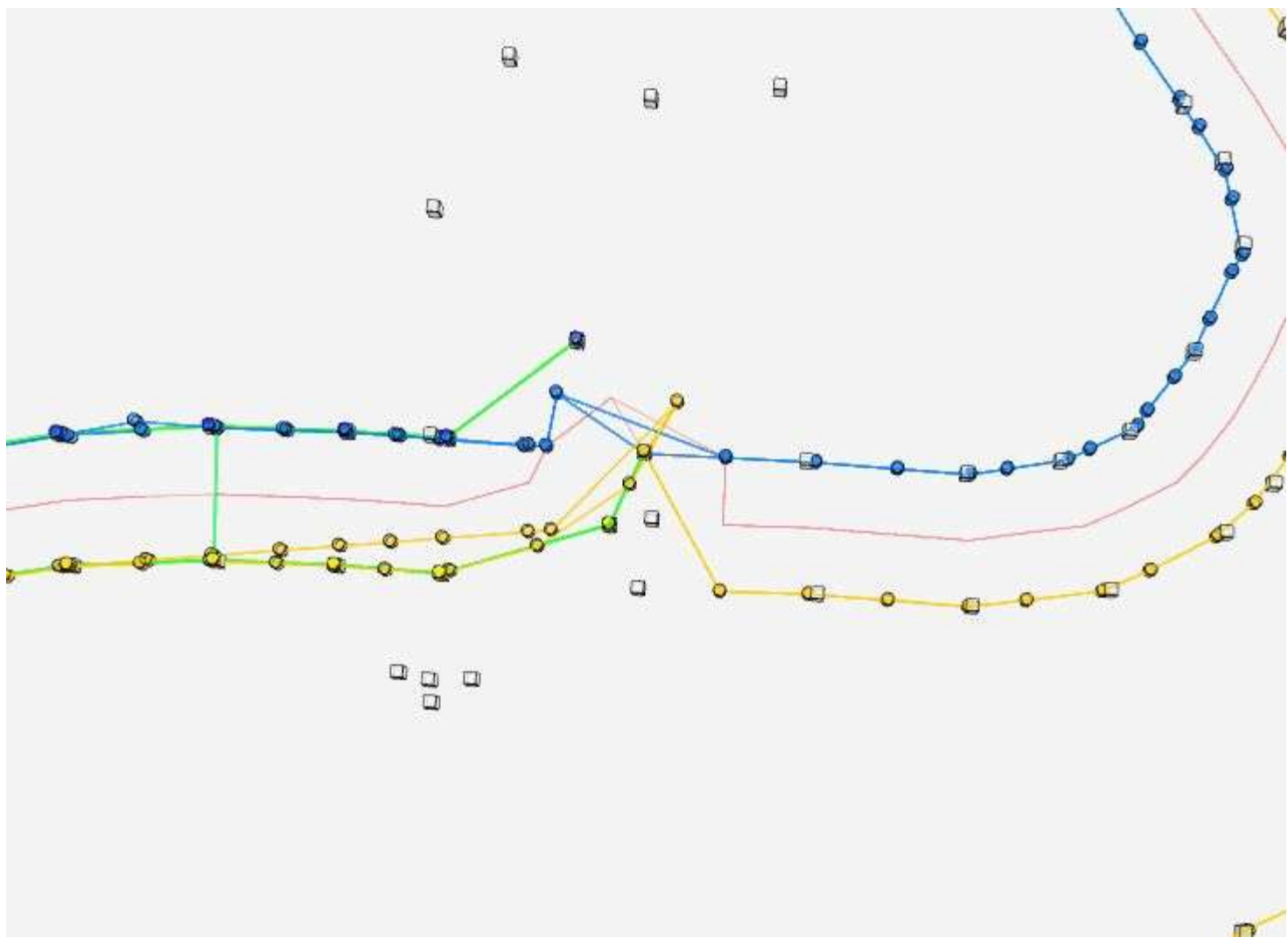


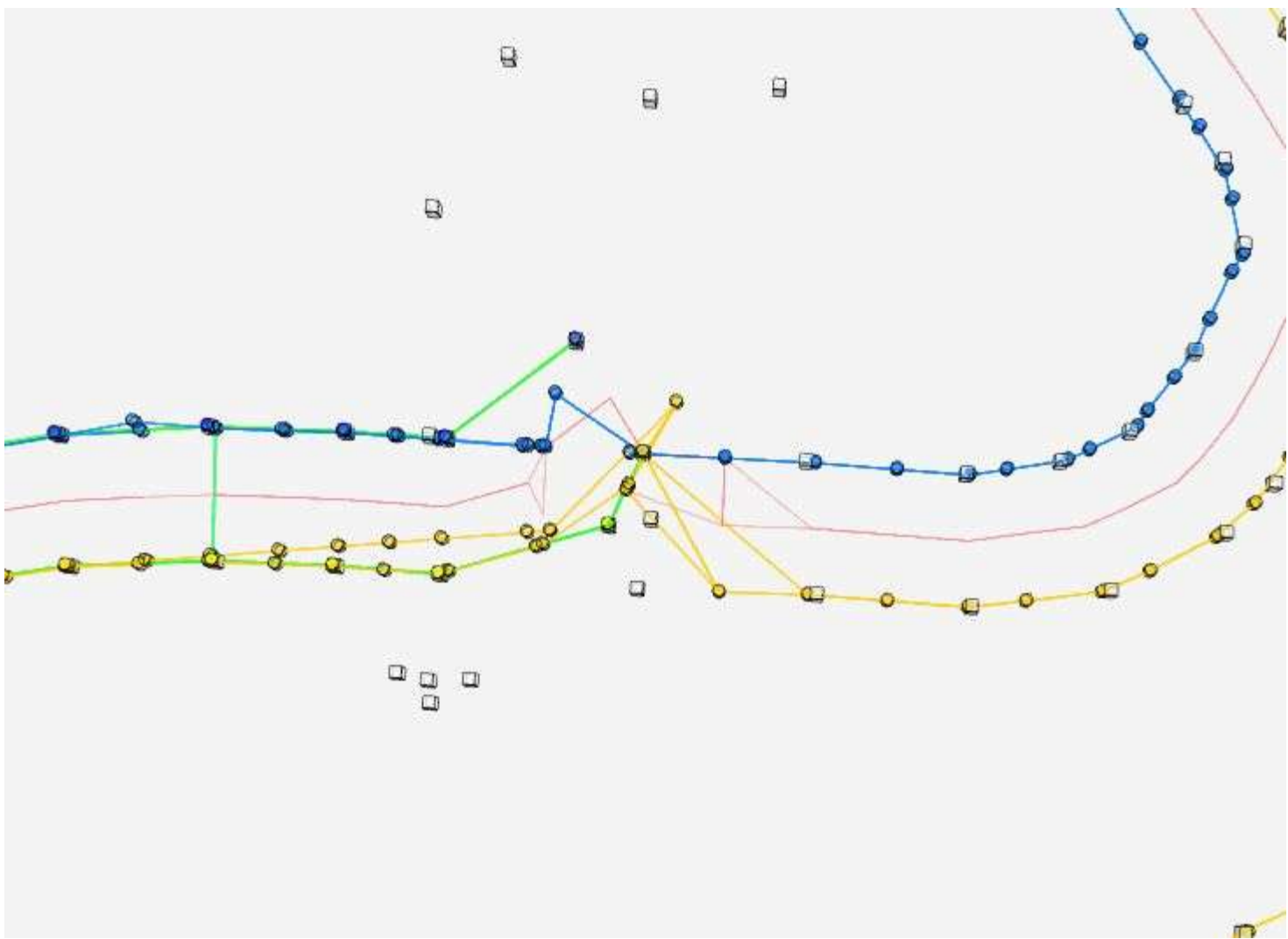


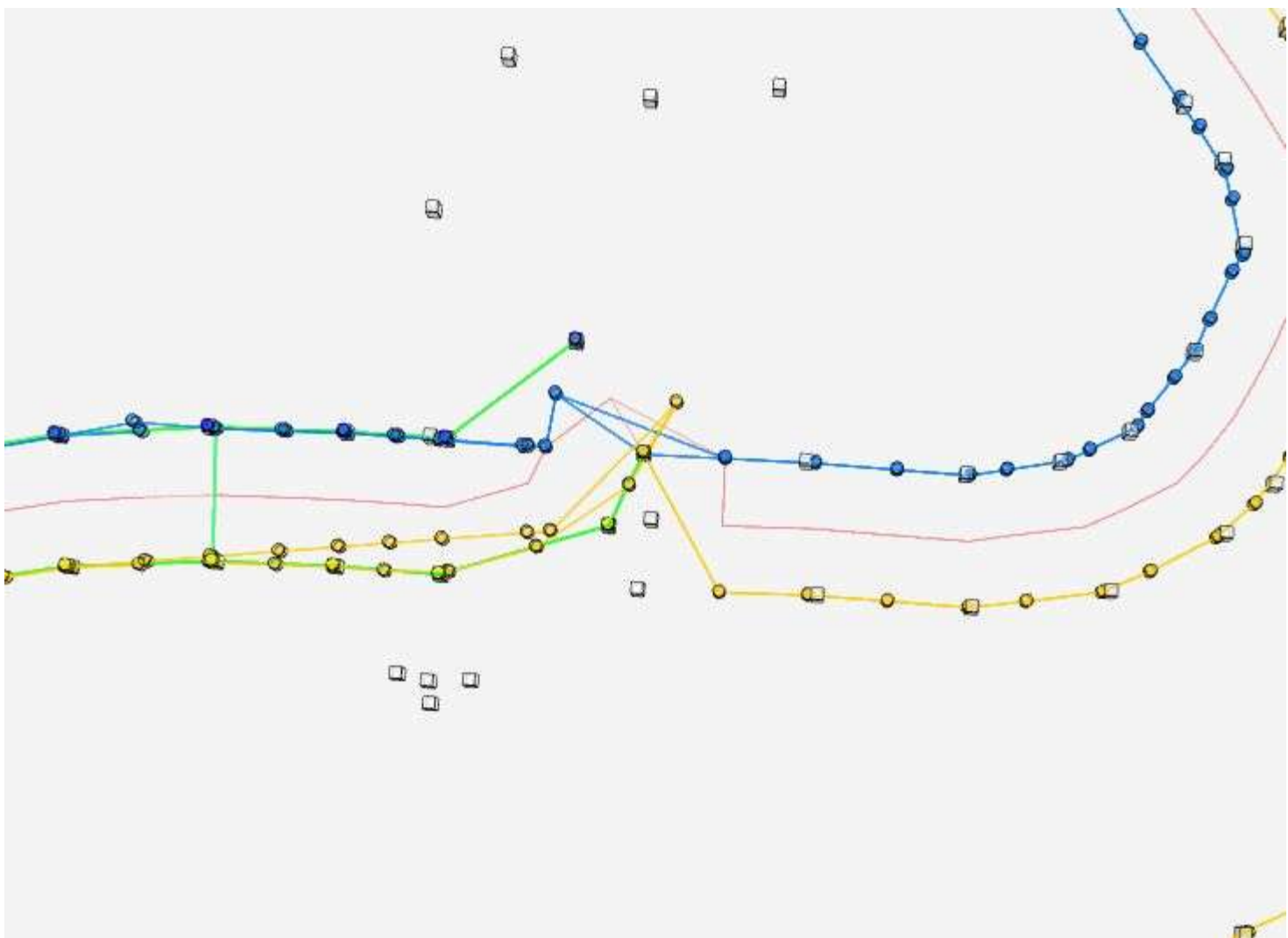


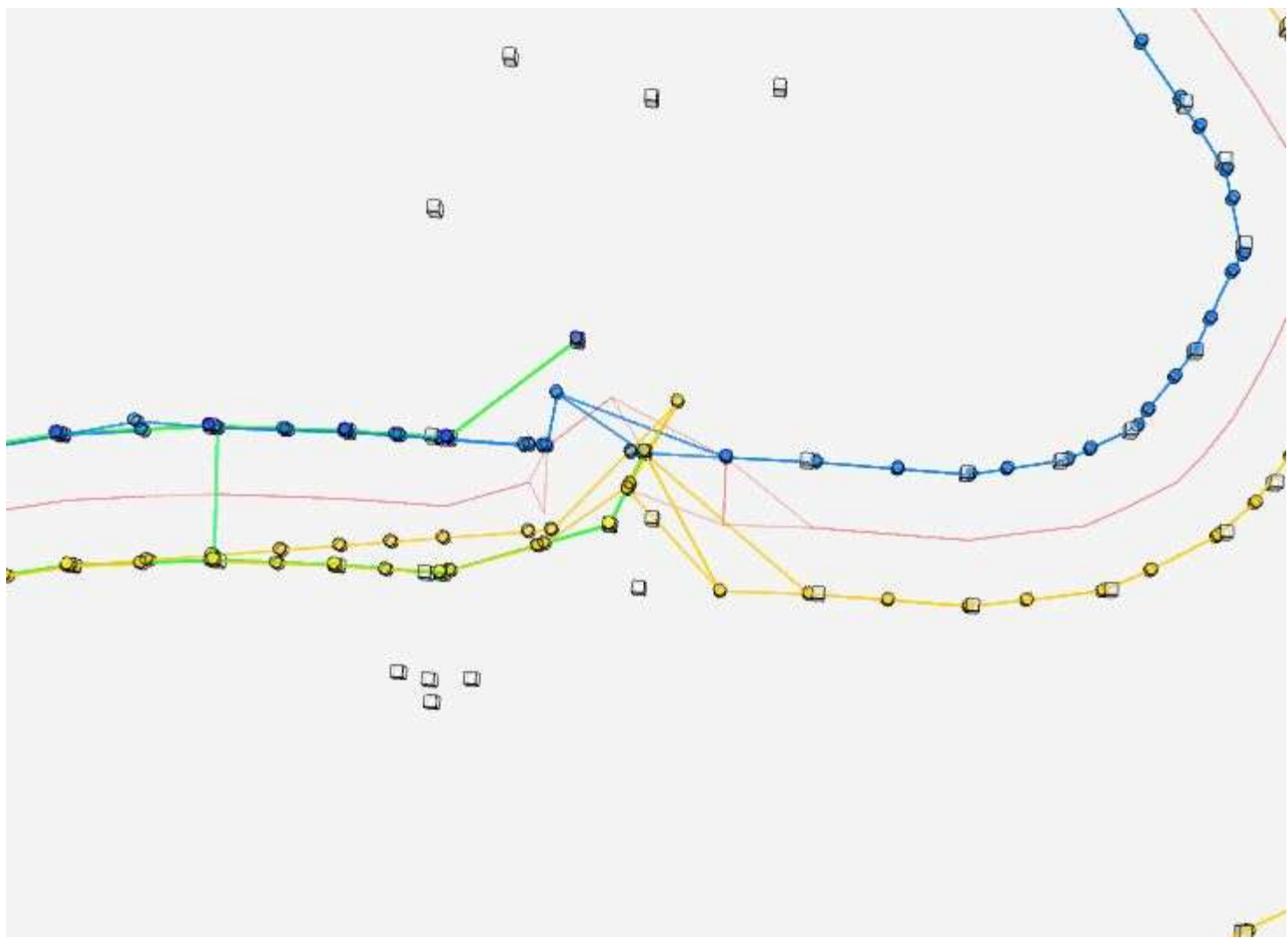


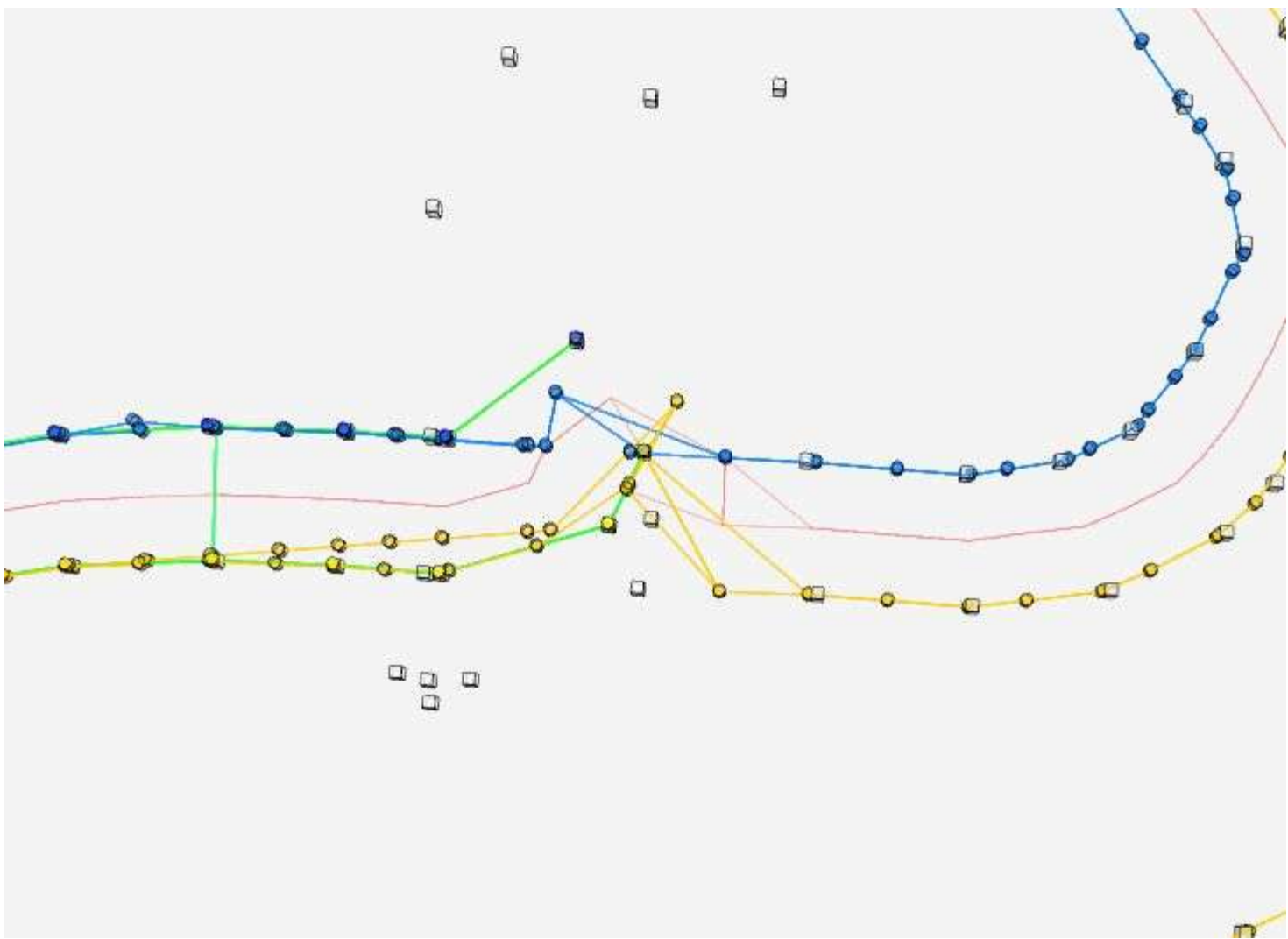


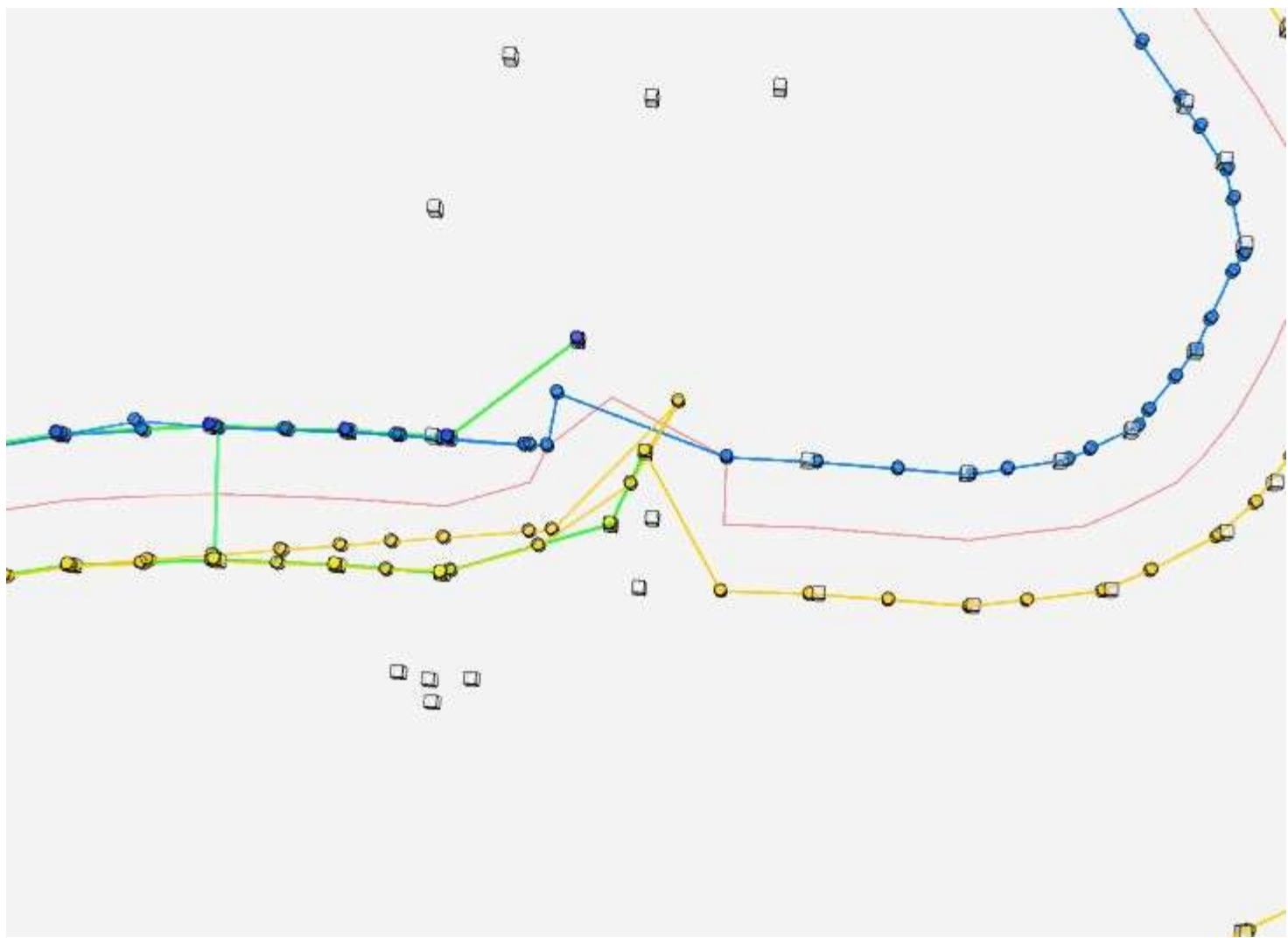


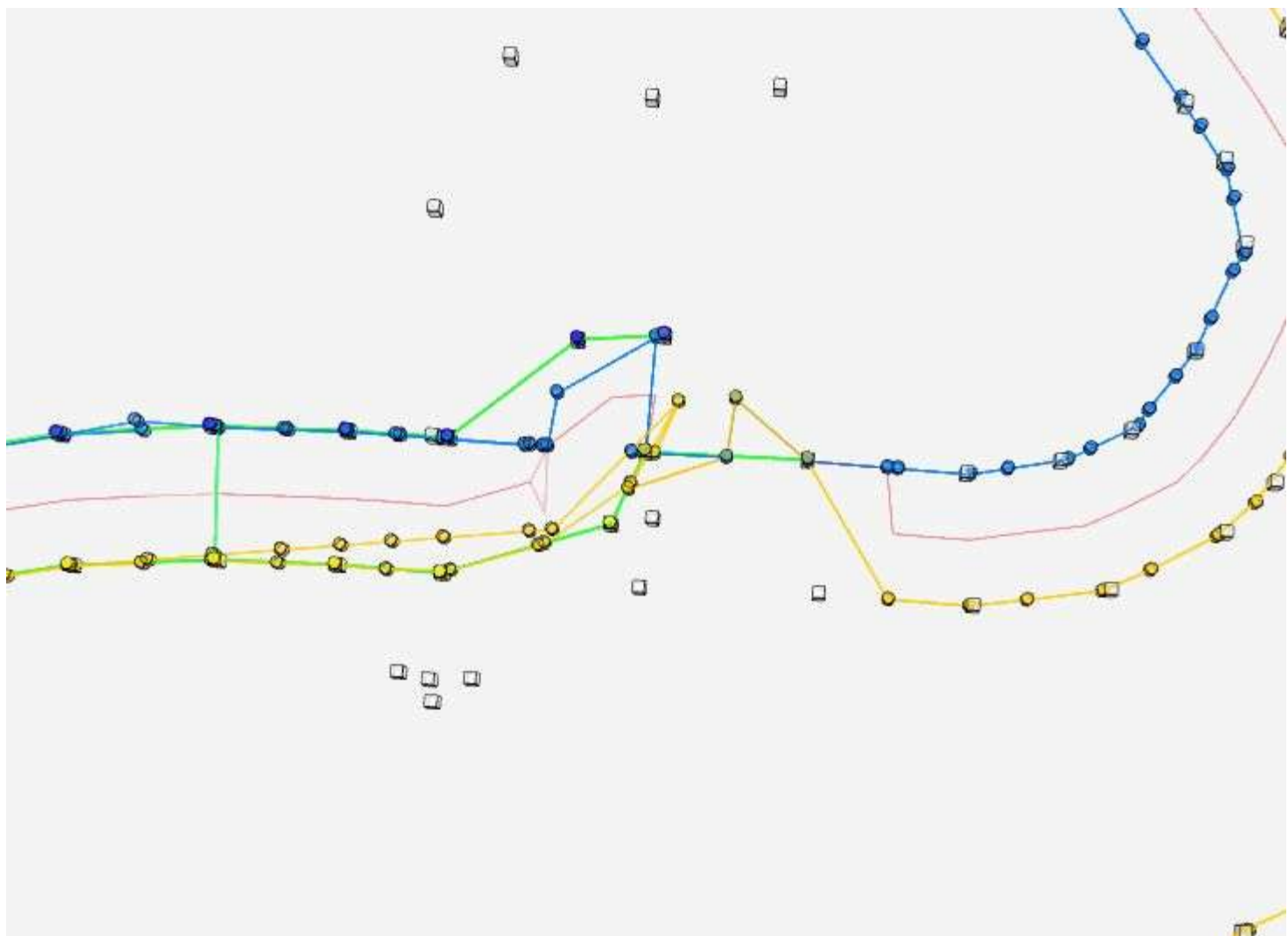


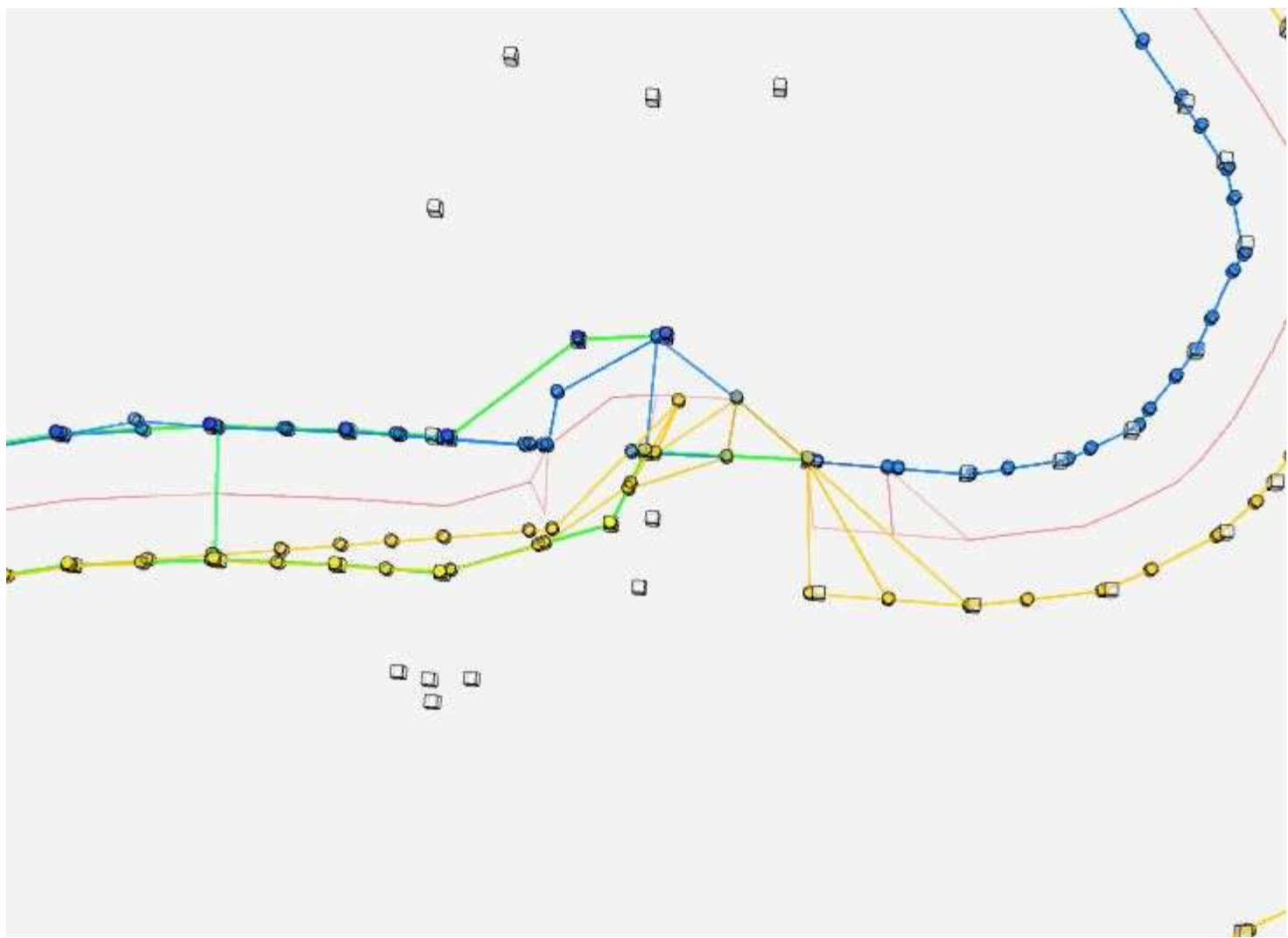


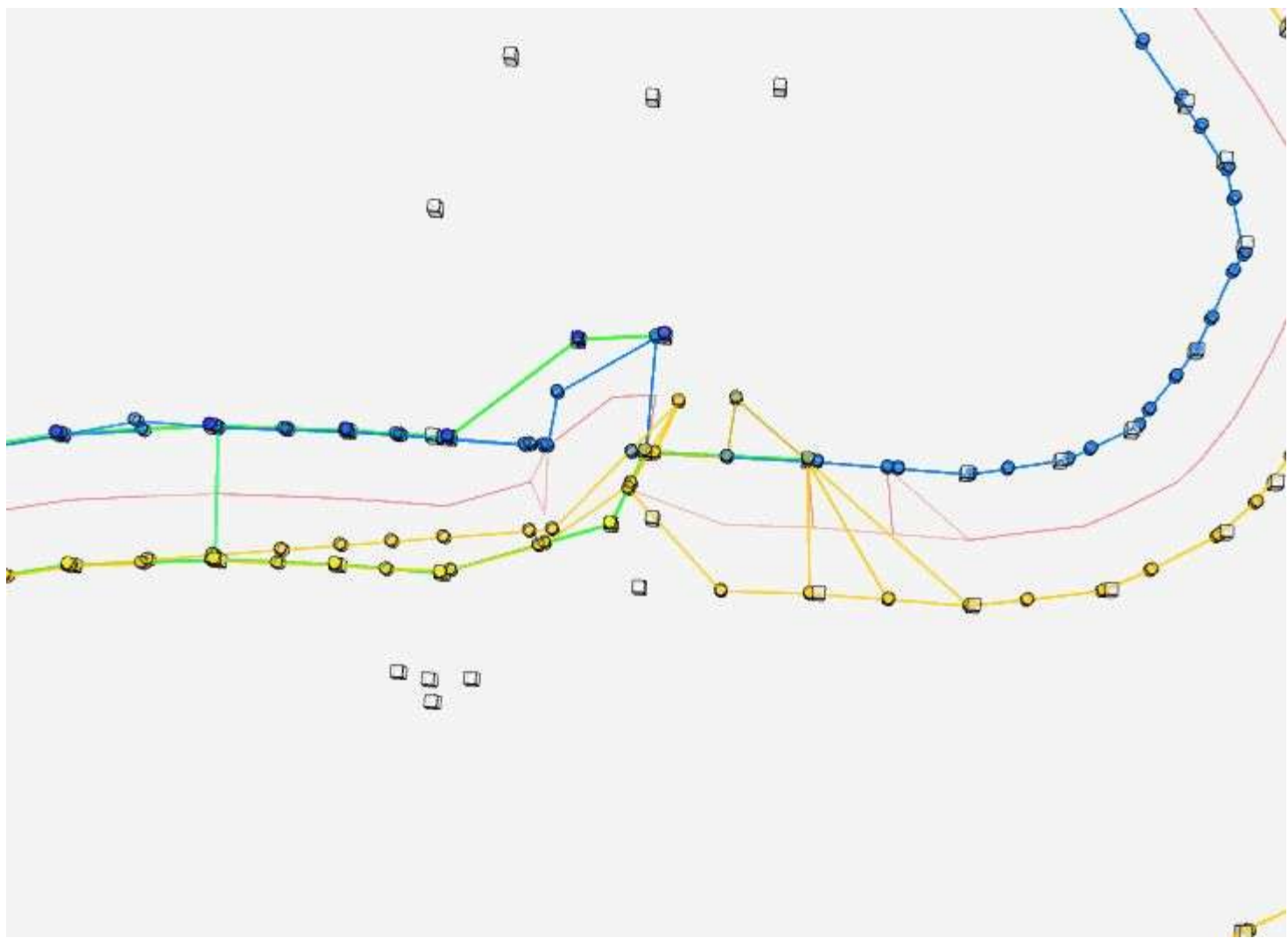


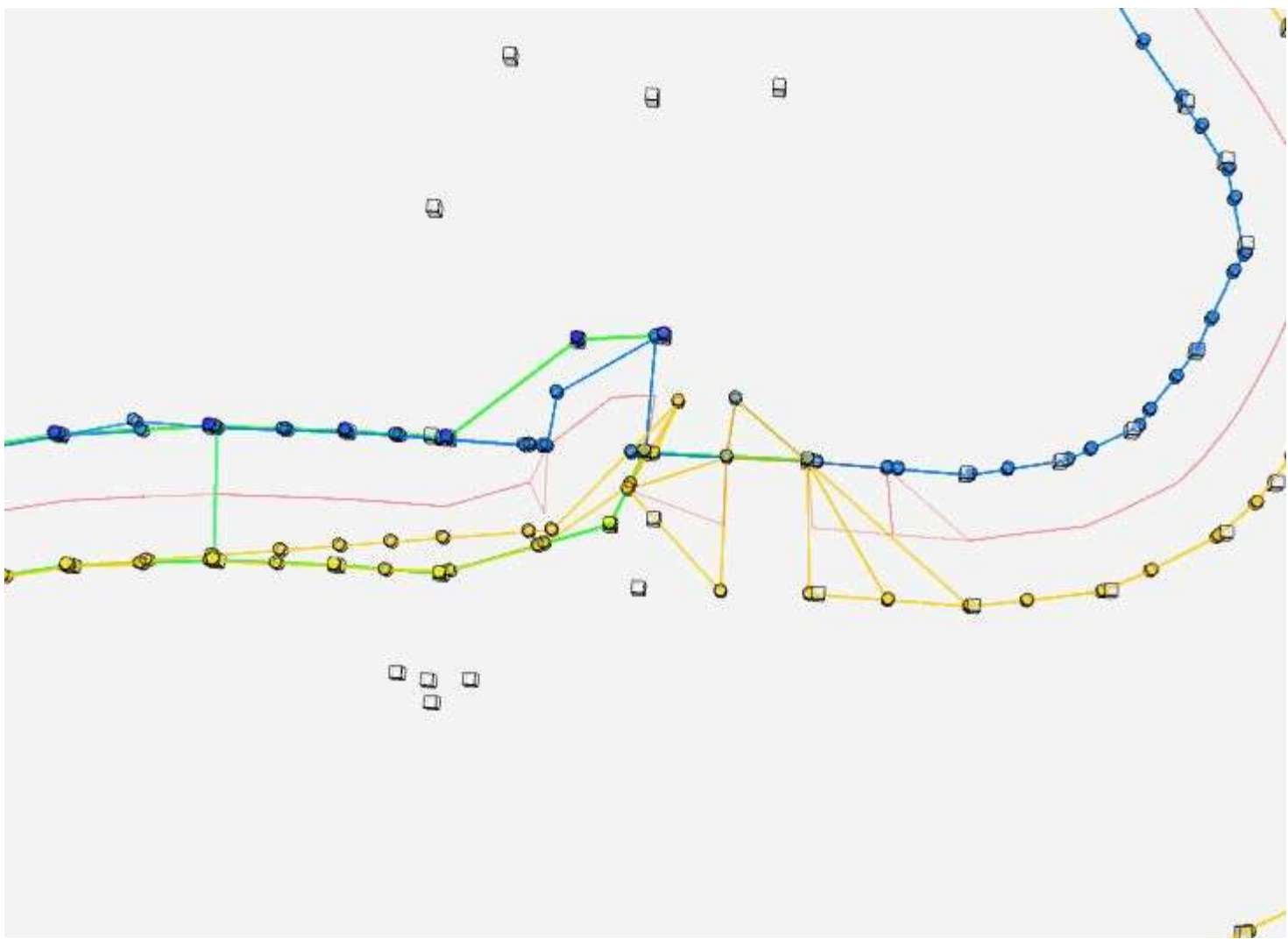


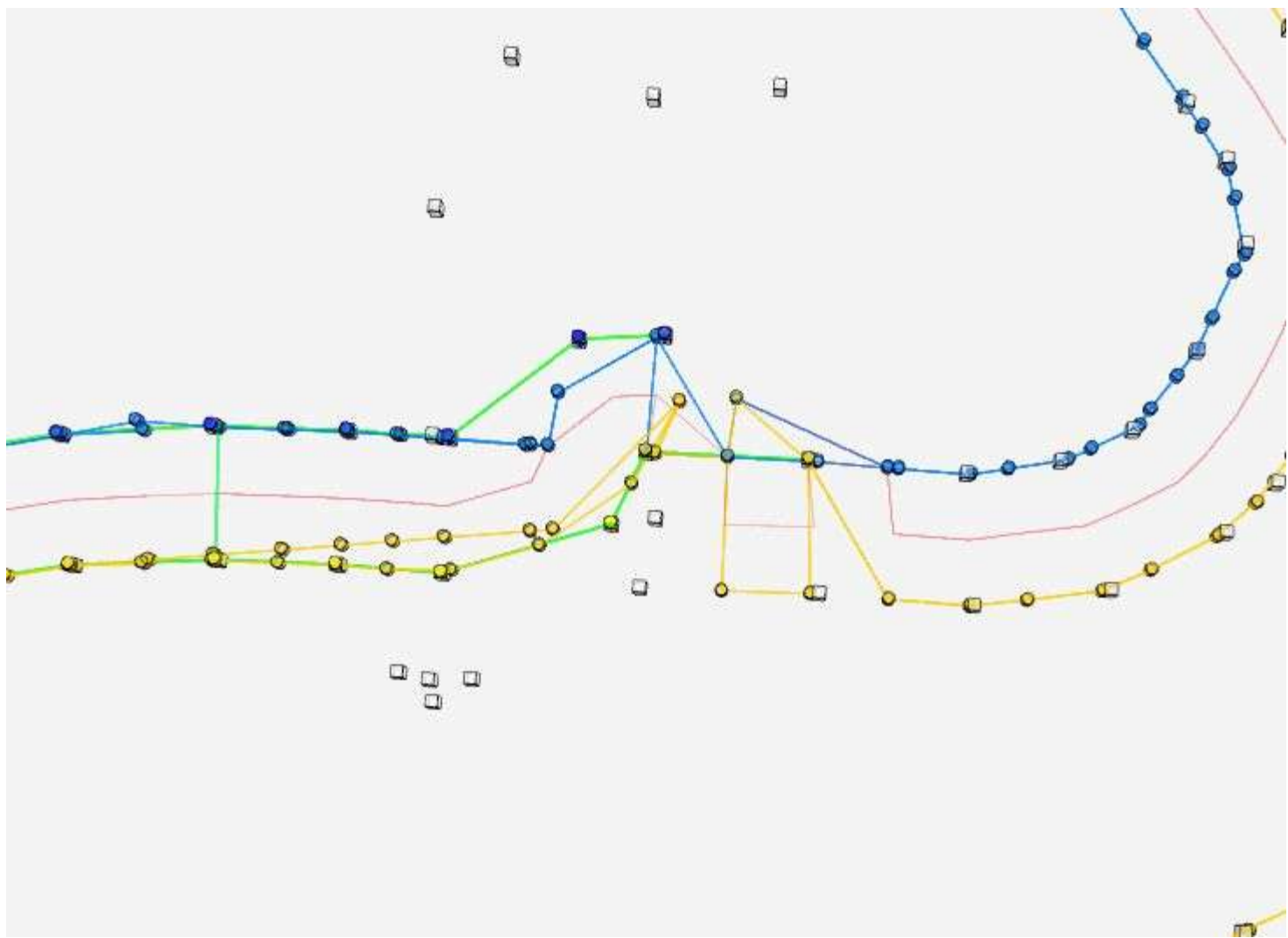


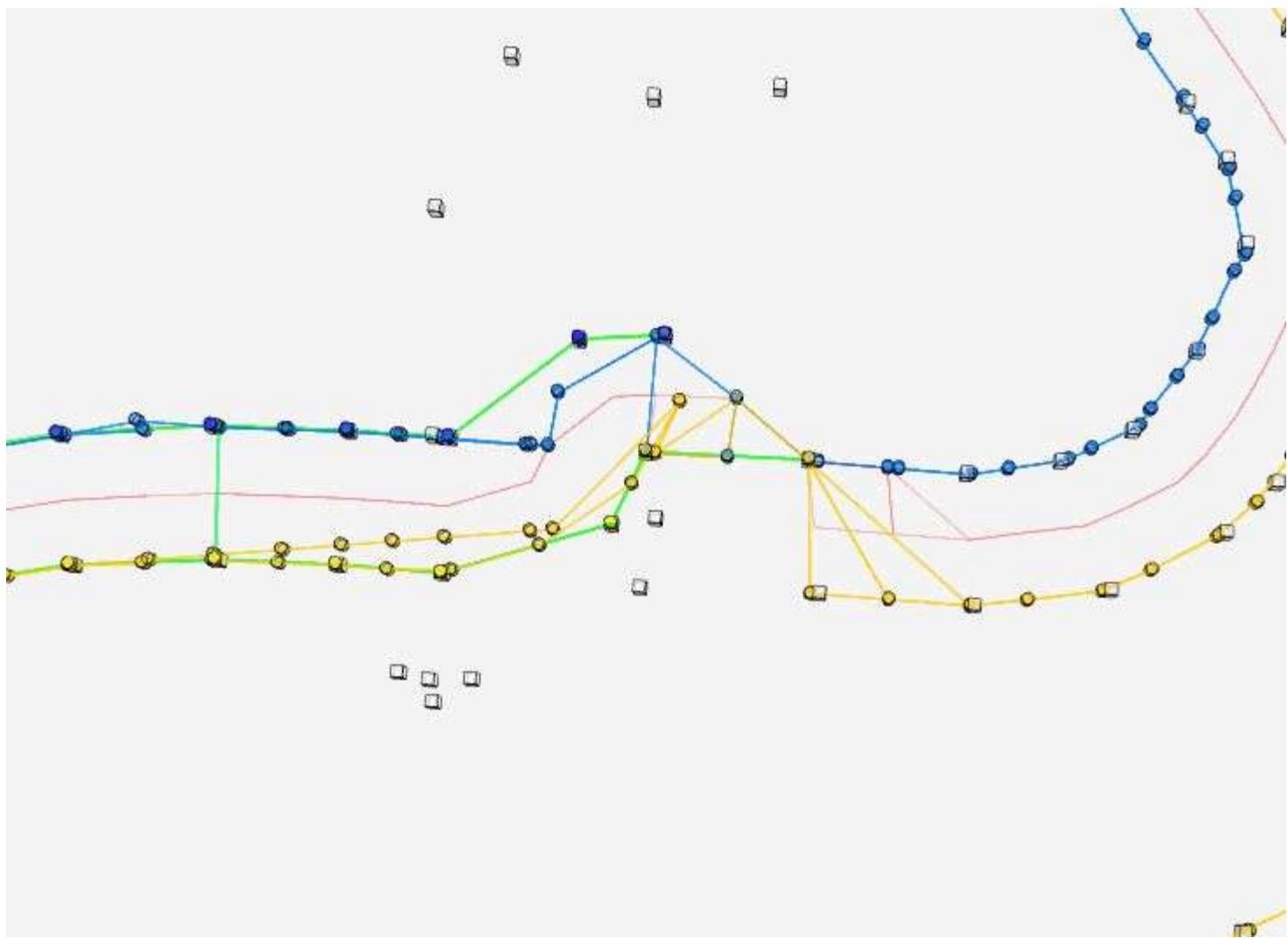


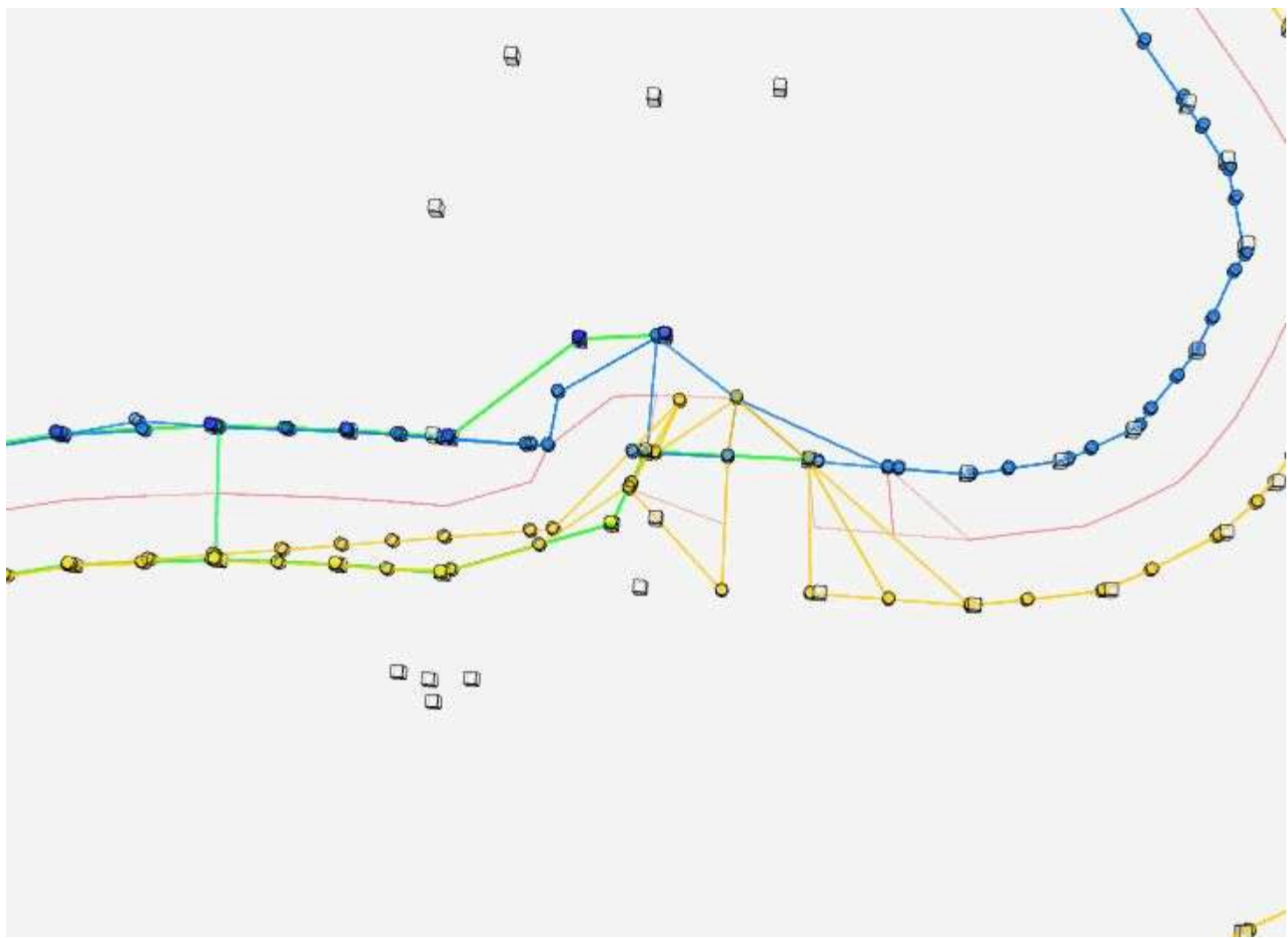


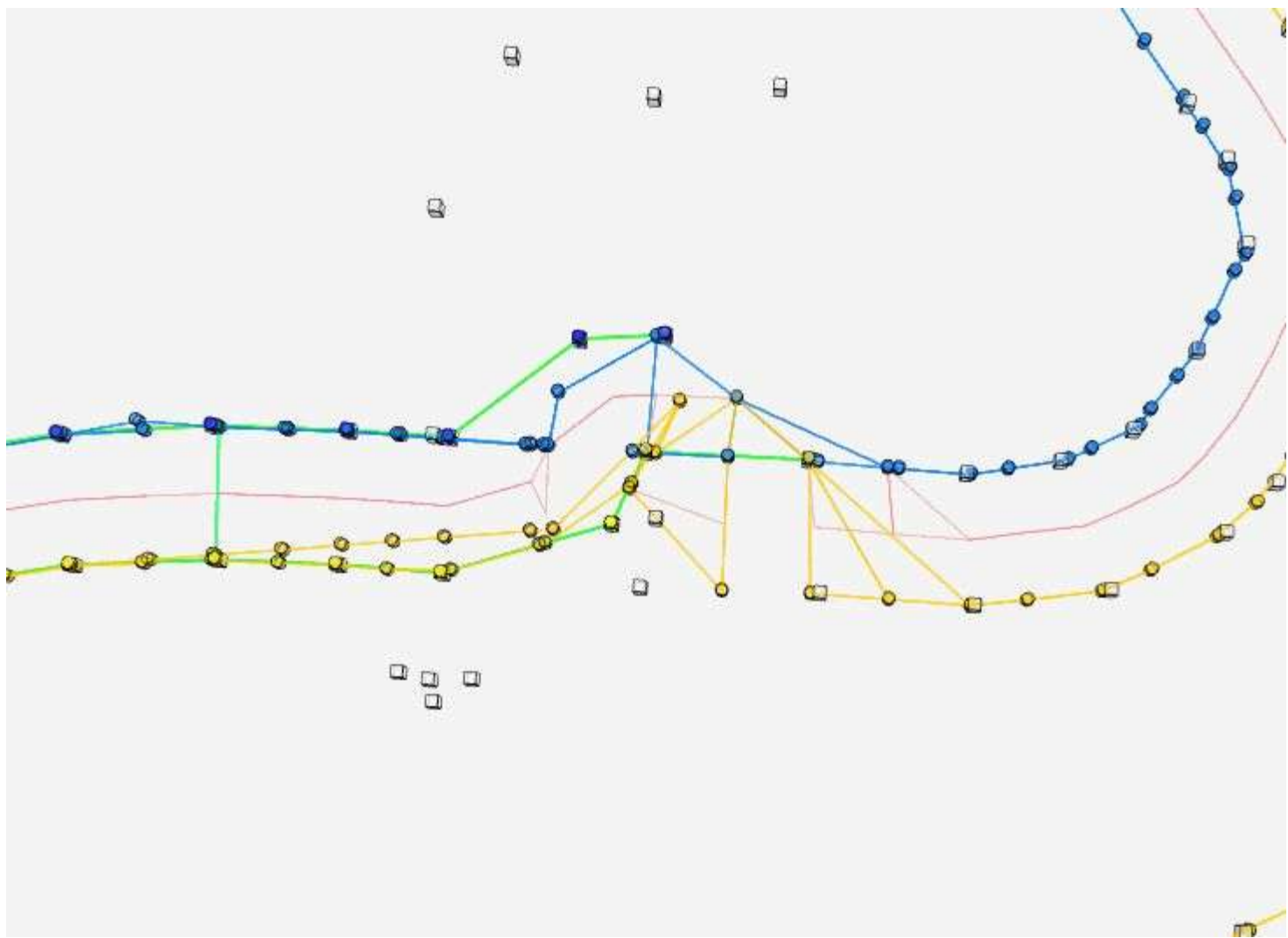


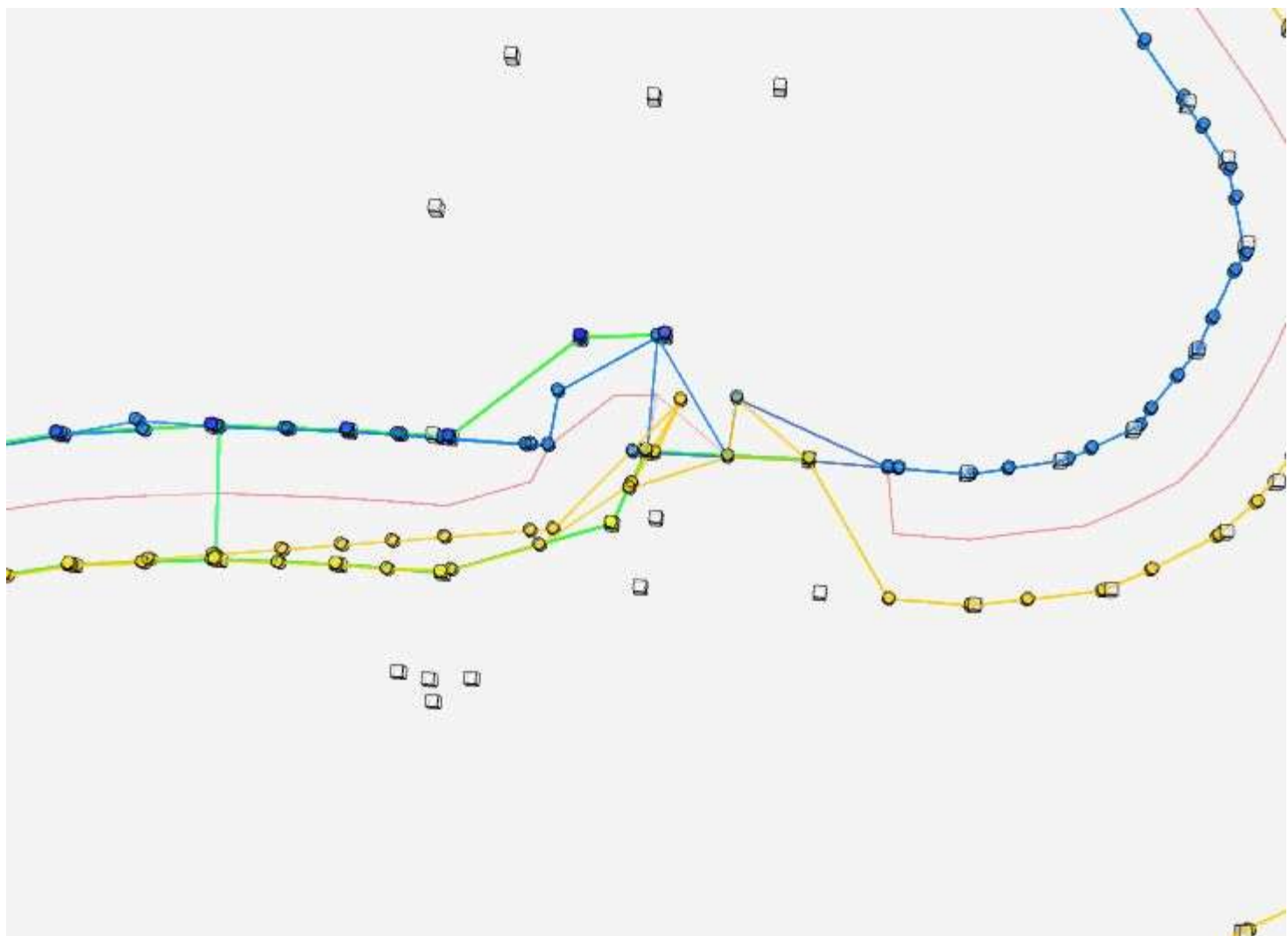


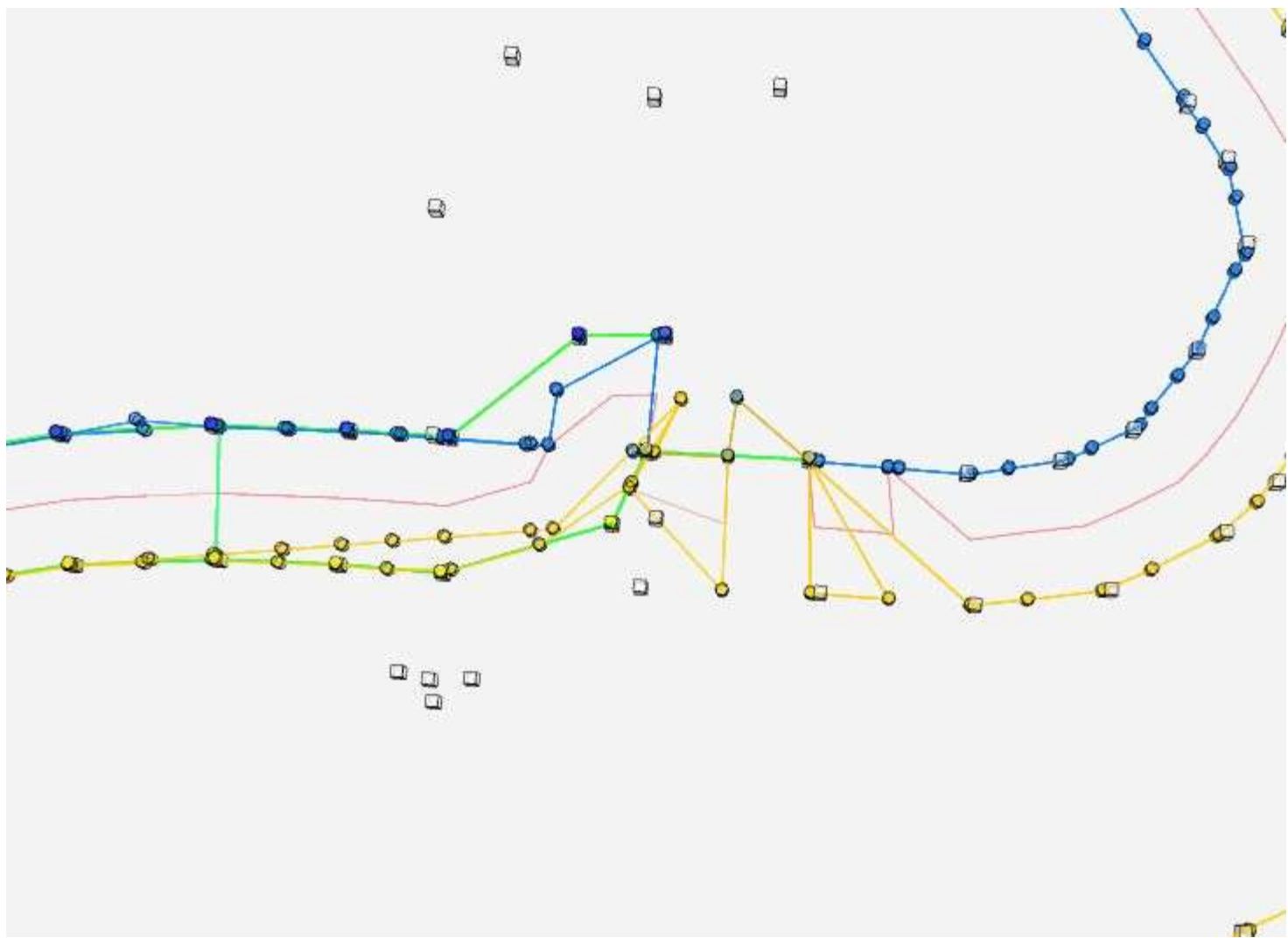


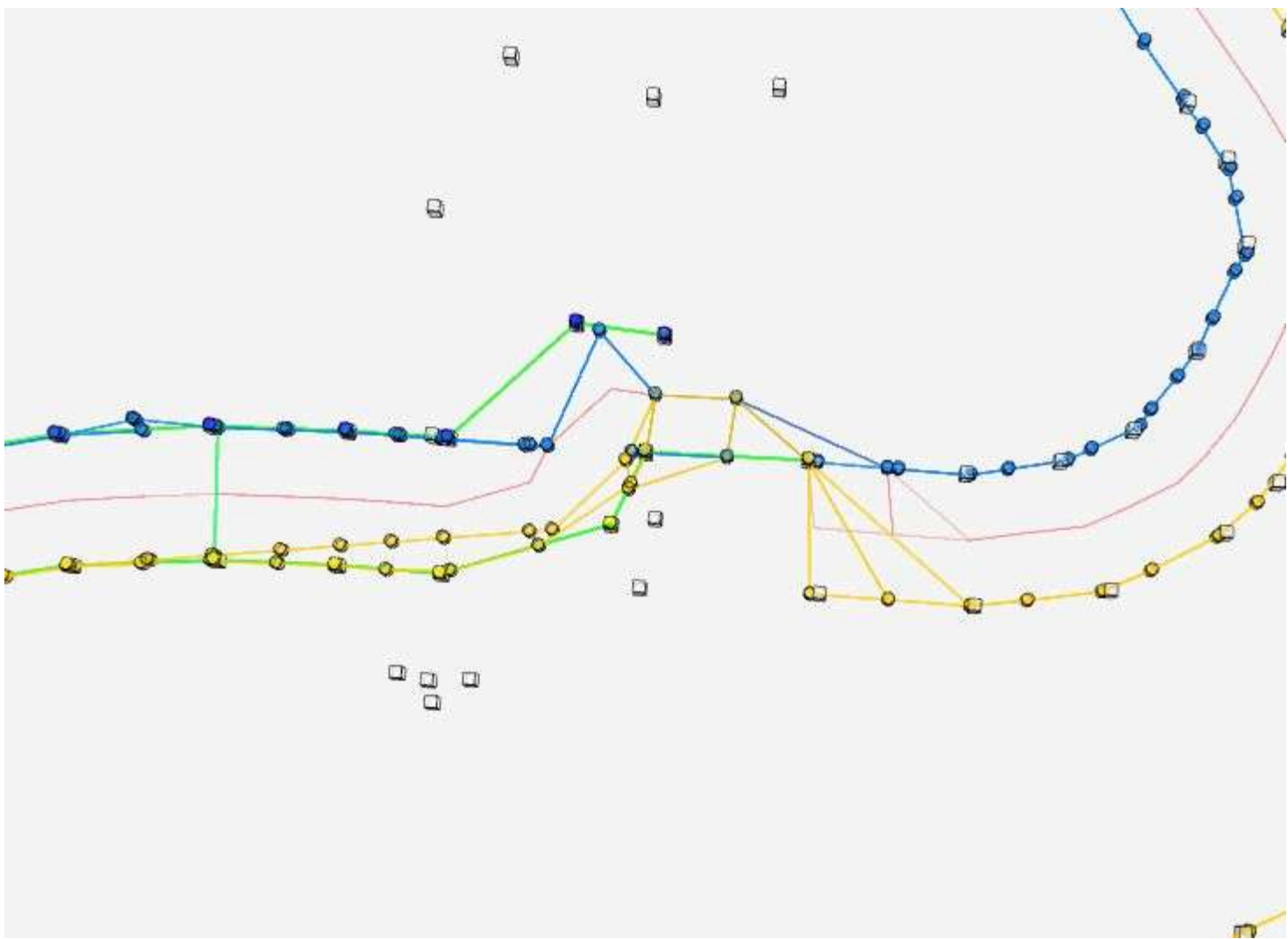


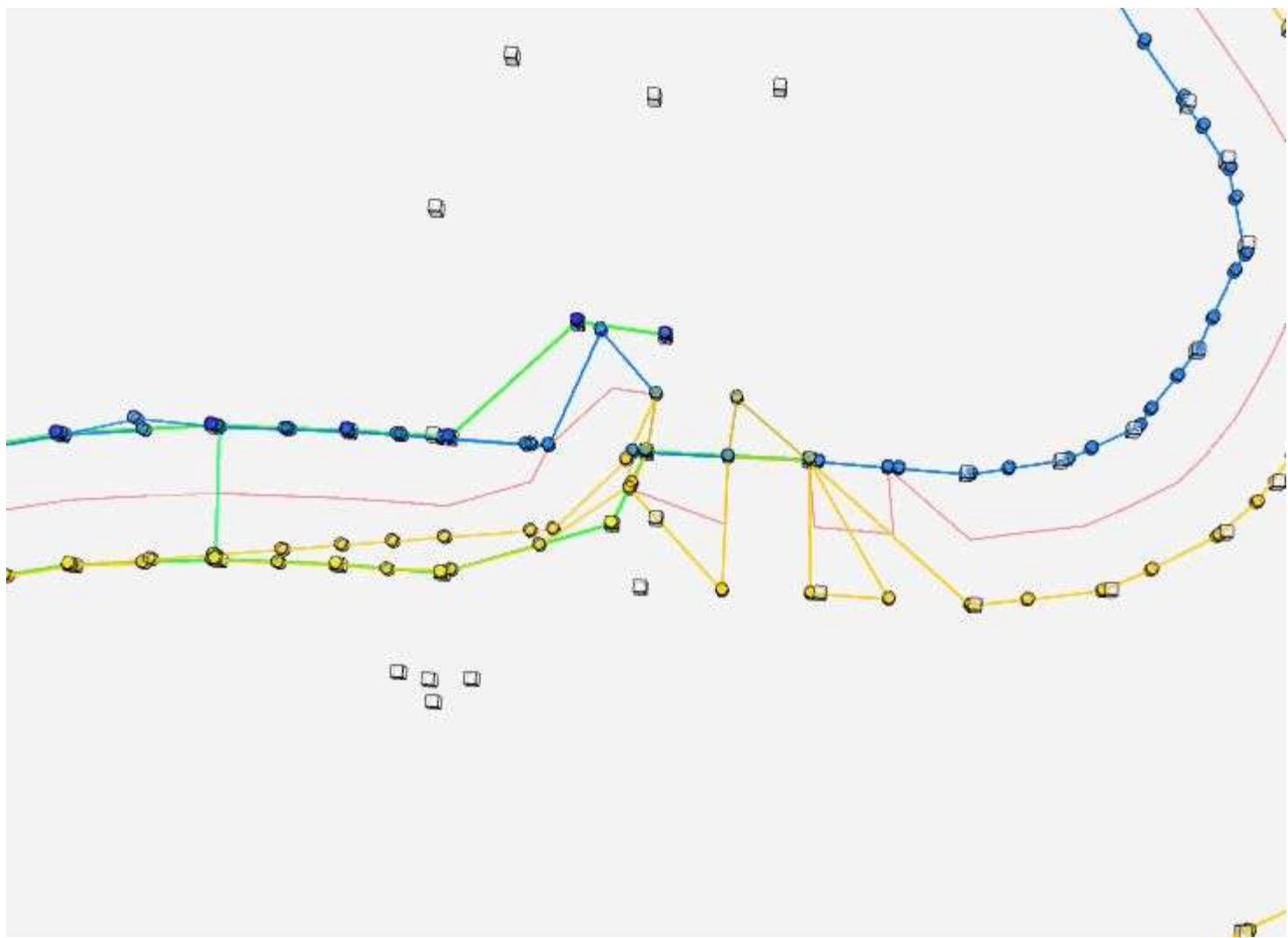


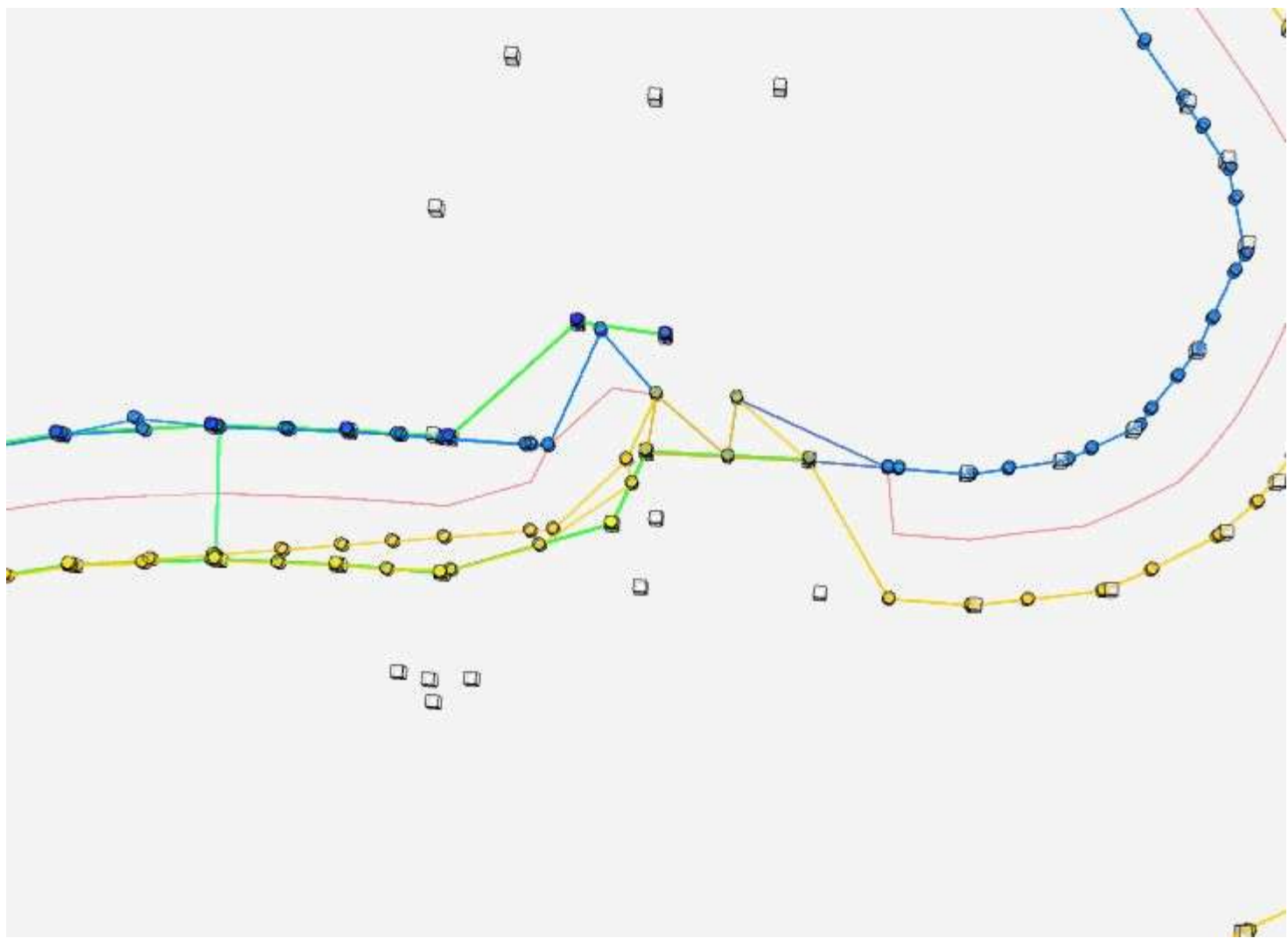


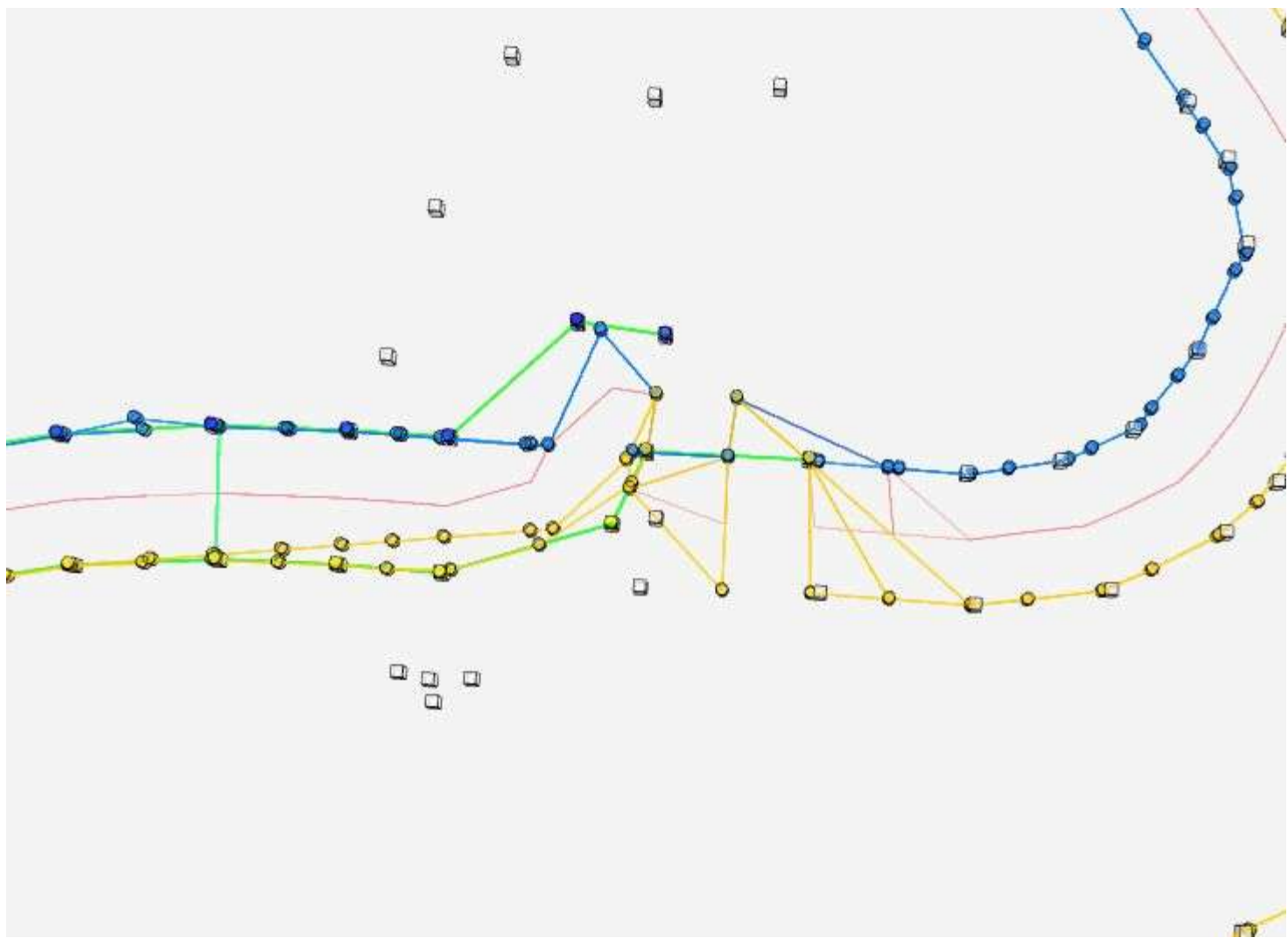


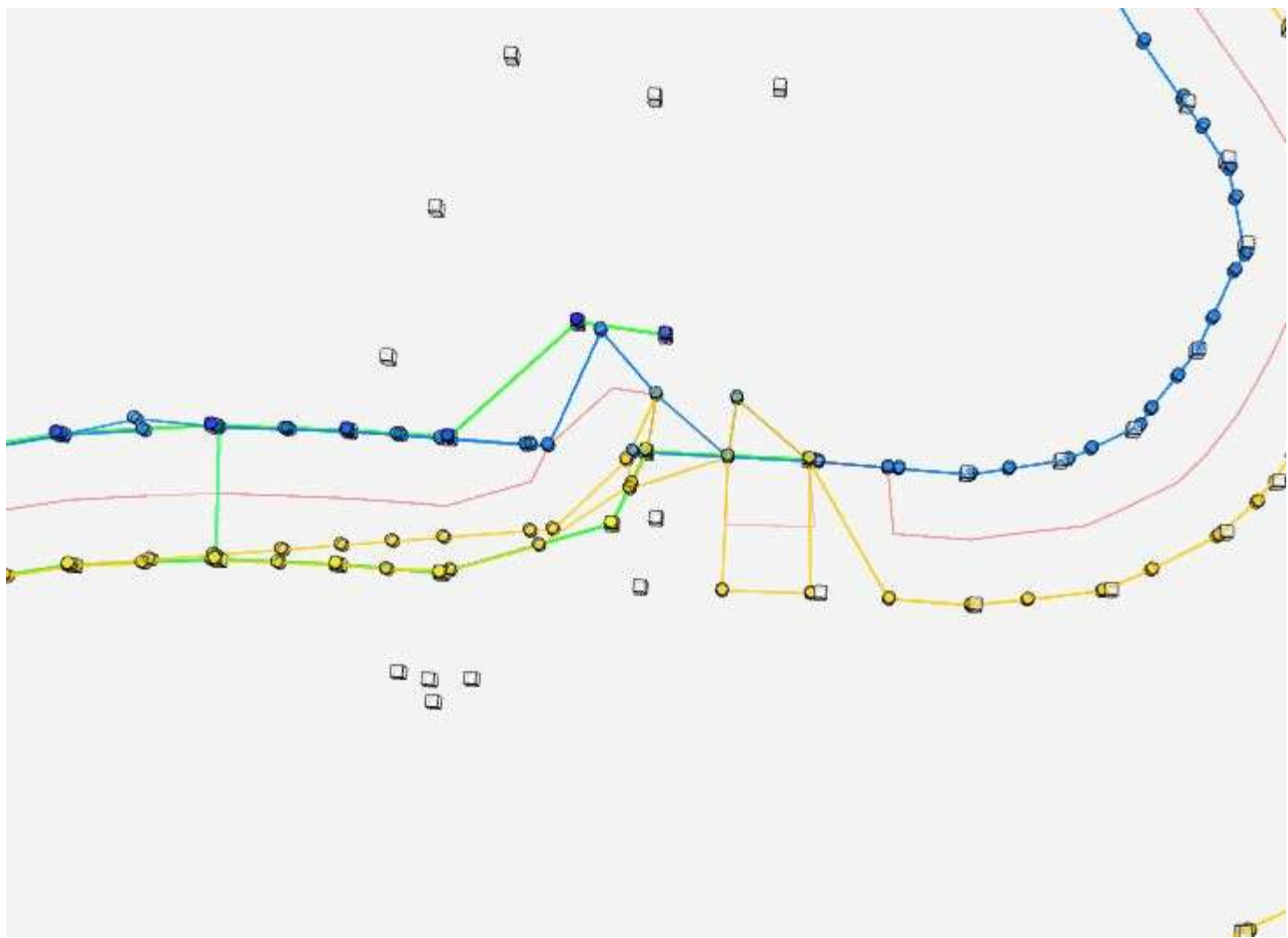


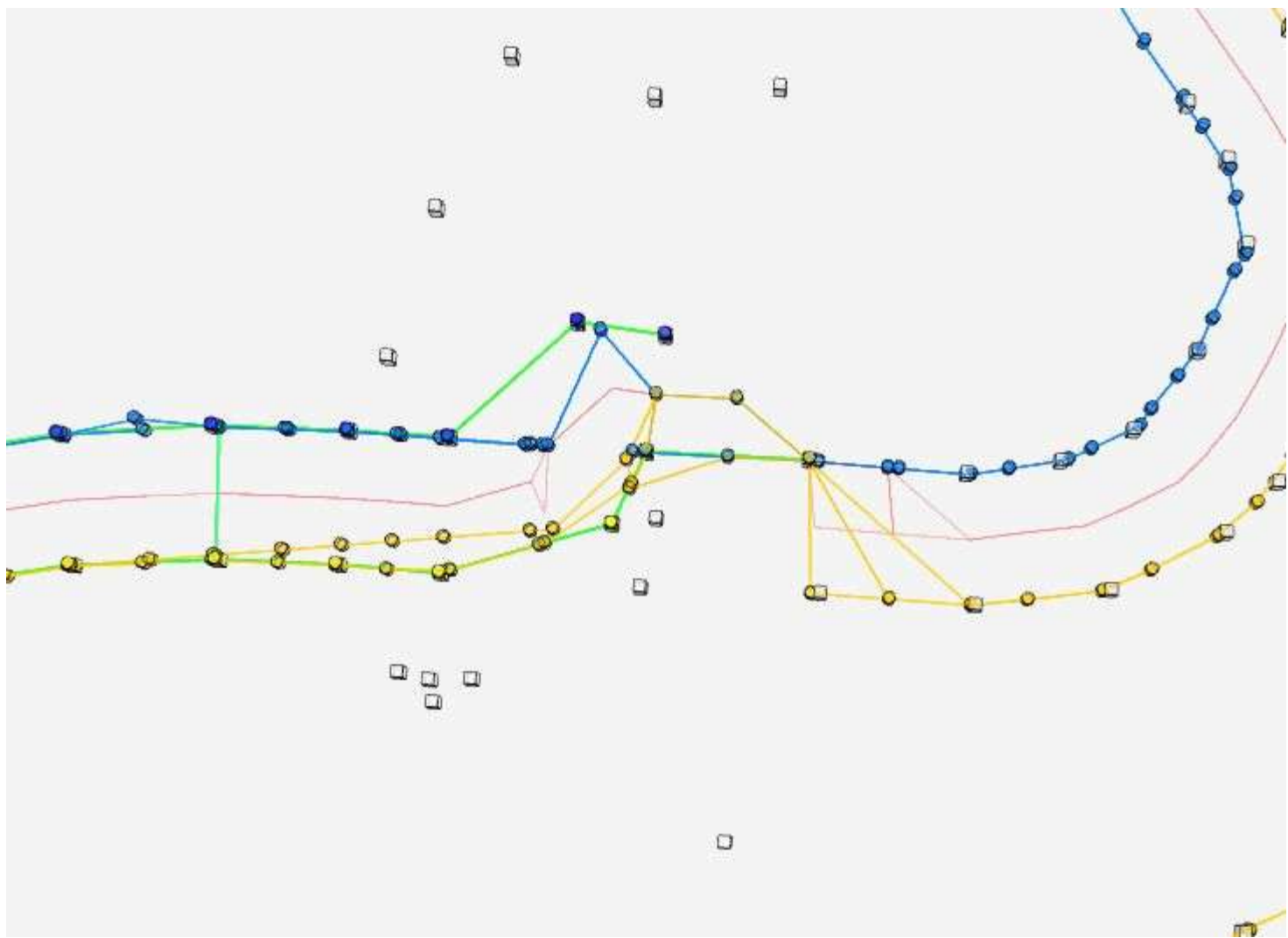




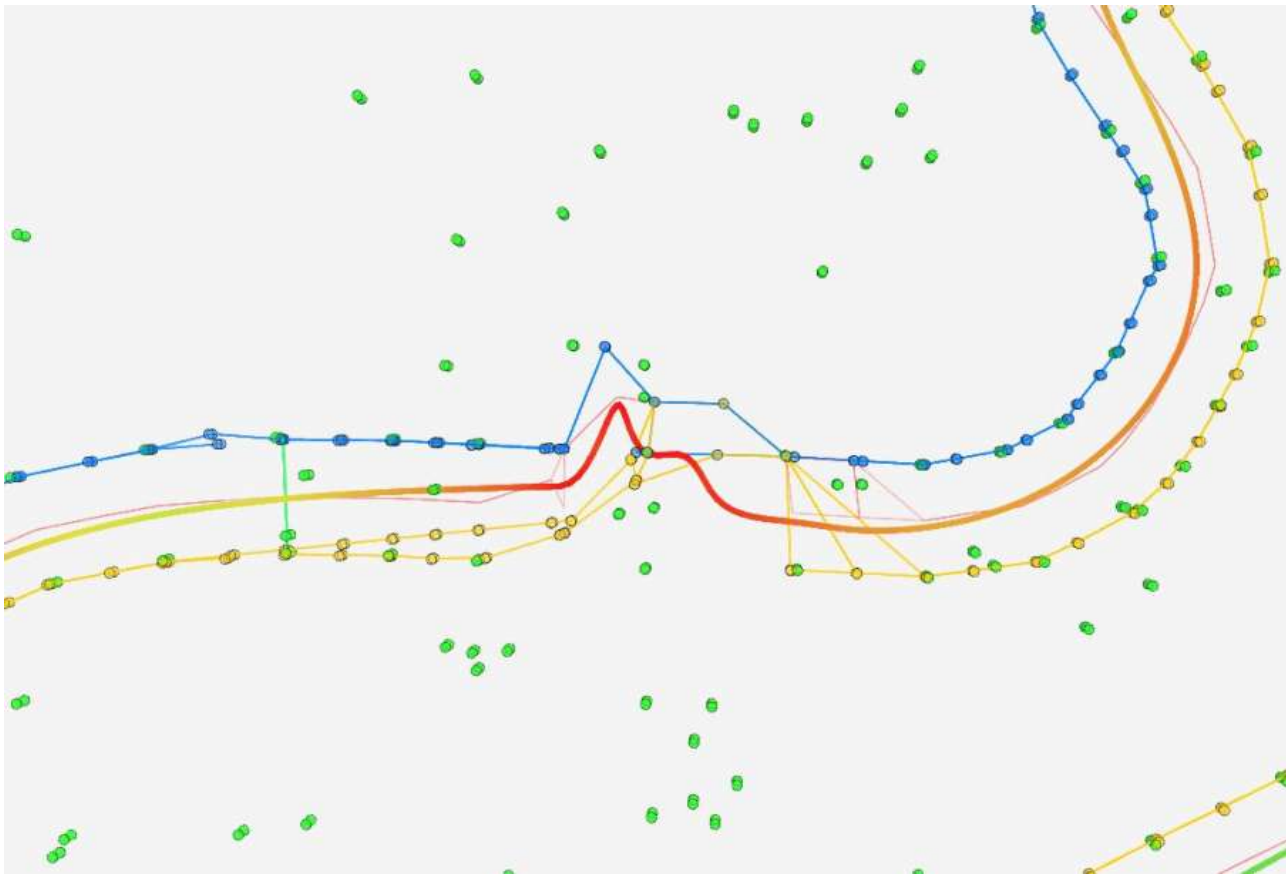








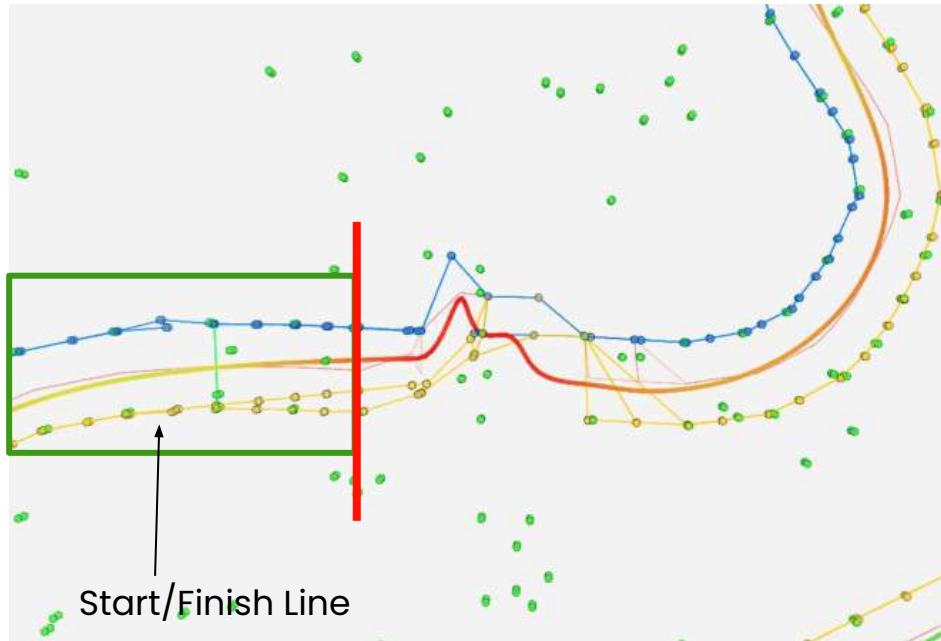
# Final product



# THE PROBLEM



after triggering the Global Path → we just **inserted the full current local path** instead of **truncating it** at the start/finish line



Path Planner:

**I'M FAST AS FUCK**

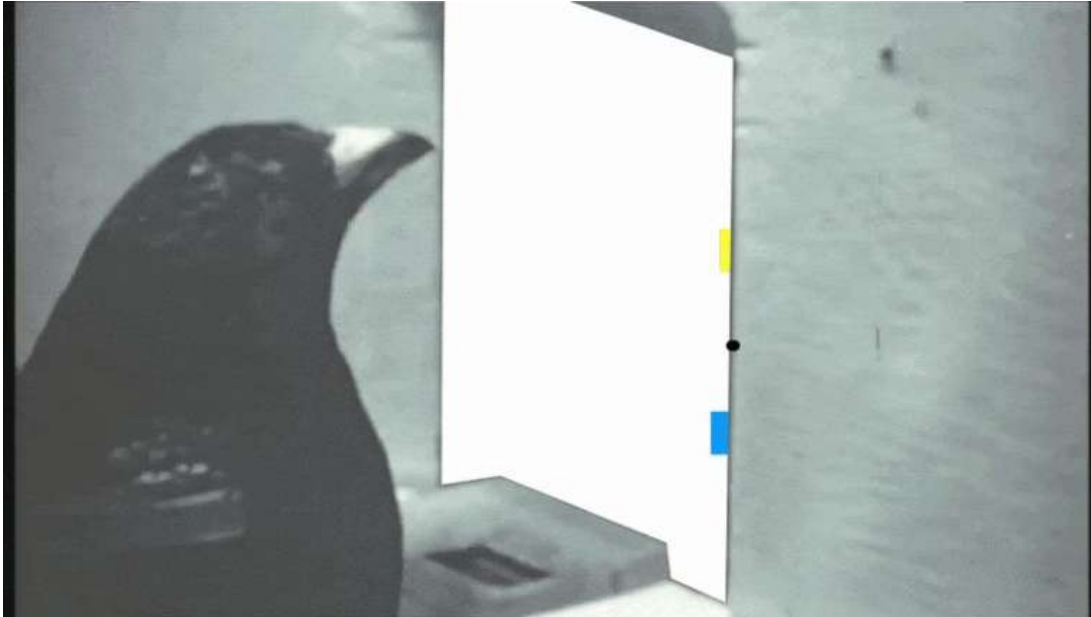




## How to solve (alternative)



- dump old path planning approach → develop PathGPT
- Machine learning (or Pigeon learning) based approach

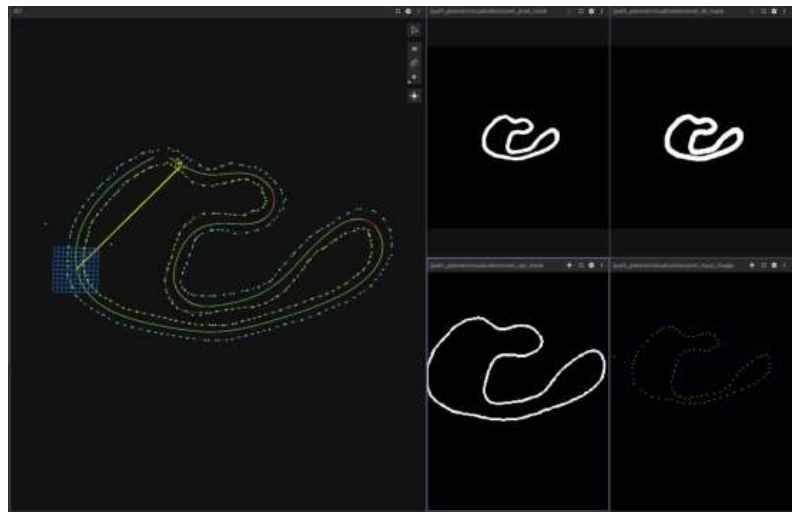




**Core idea:** Convert cone detections into a top-down binary image

**Model:** U-Net semantic segmentation to classify each pixel as track / non-track

- Color information gives a slight improvement
- Model is still sensitive to false positives and especially consecutive false negatives



# FSG25 Special Award



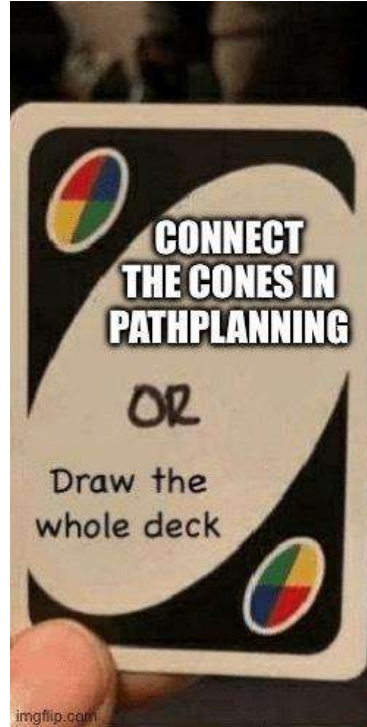
our offtrack performance



DV offtrack performance

our Path Planning  
guy Lukas <3 (he got fired)

# Memes



our car name

# Feedback QR Code



Thank you for your attention !

See you at FSG !

Give us feedback <3

Here are all our slides:

[www.elbflorace.de/publications](http://www.elbflorace.de/publications)

