Yandex

Self Driving Car Course

Seminar on Graph SLAM

Dmitri Kovalenko

2021.04.02

This is a coding seminar

```
# will need:
python3,
pip (or it’s alternative)

cd …/seminar02-graph-slam/code
python3 -m pip install -r ./requirements.txt
```

Handouts:

```
cd …/seminar02-graph-slam/materials; ls
homework.pdf  seminar.pdf
seminar_slides.pdf  theory.pdf
```

This is a coding seminar

4 coding tasks for the semitar,

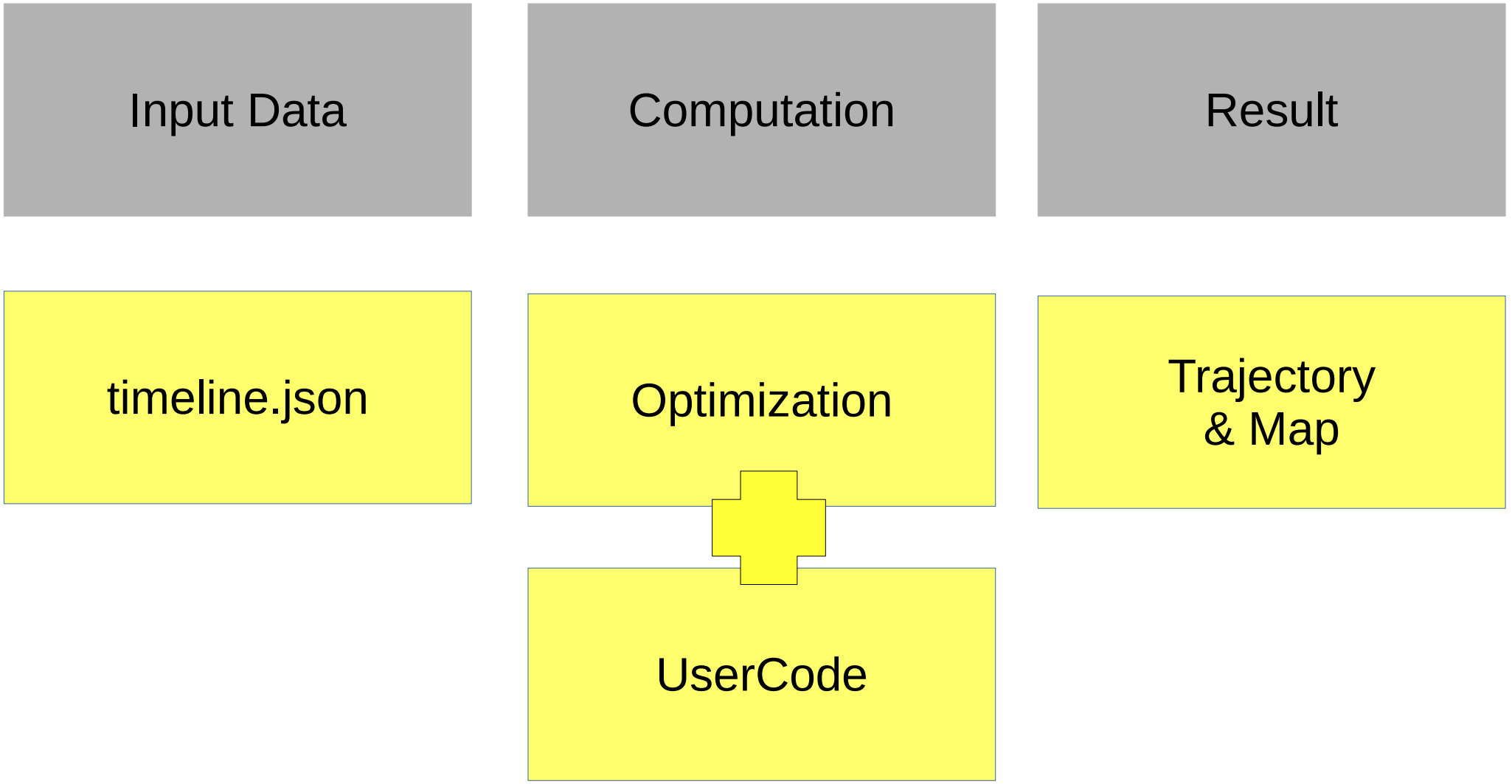
with perceived complexity:

- 30% 15% 50% 5%

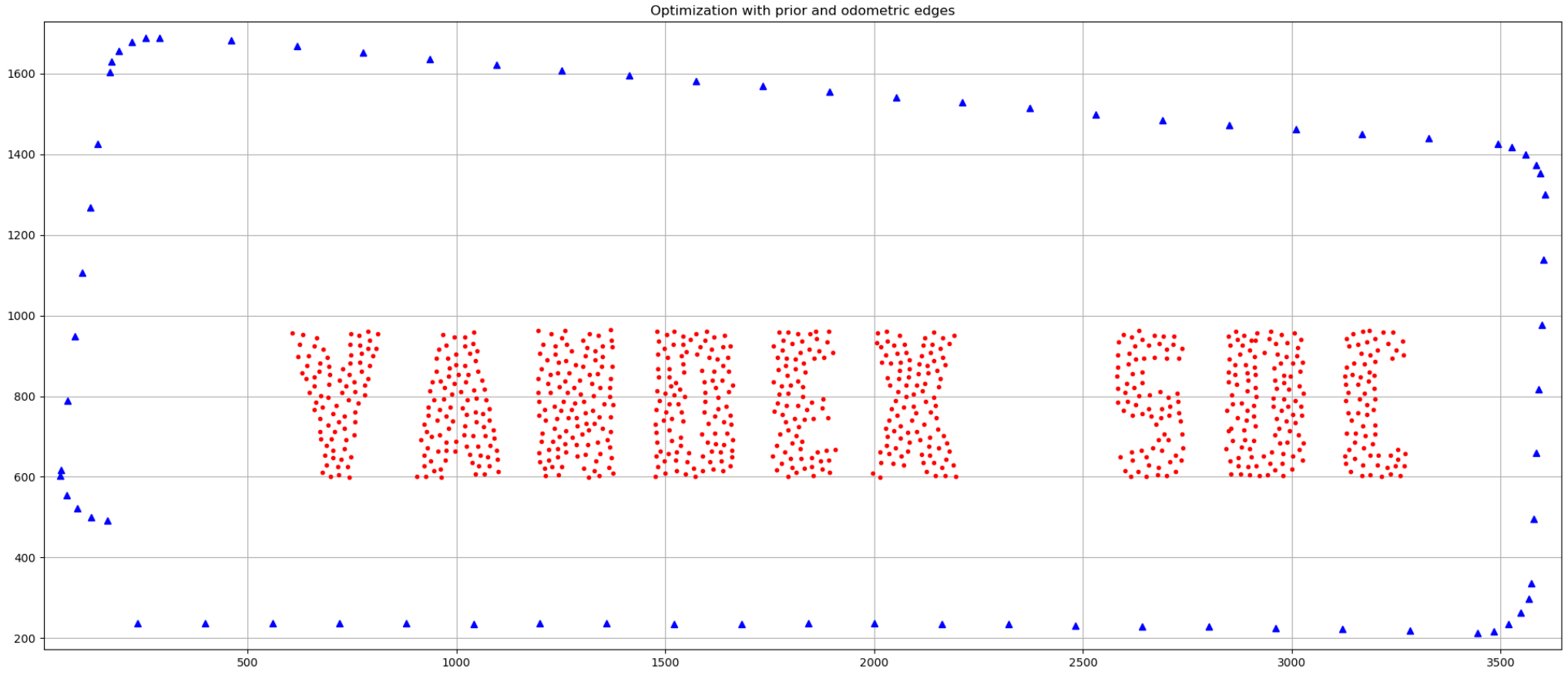
2 coding tasks for the homework

- 10% 20%

About the problem



Result


Optimization with prior and odometric edges
1600
1400
1200
1000
800
600
400
200
500
1000
1500
2000
2500
3000
3500

Input Data (timeline.json)

```
[
[...],  # list of events at timestamp 0
[...],  # list of events at timestamp 1, etc
]
```

Seminar.Task#1
Refer to: theory.pdf, Sec. “Data”

Event types:

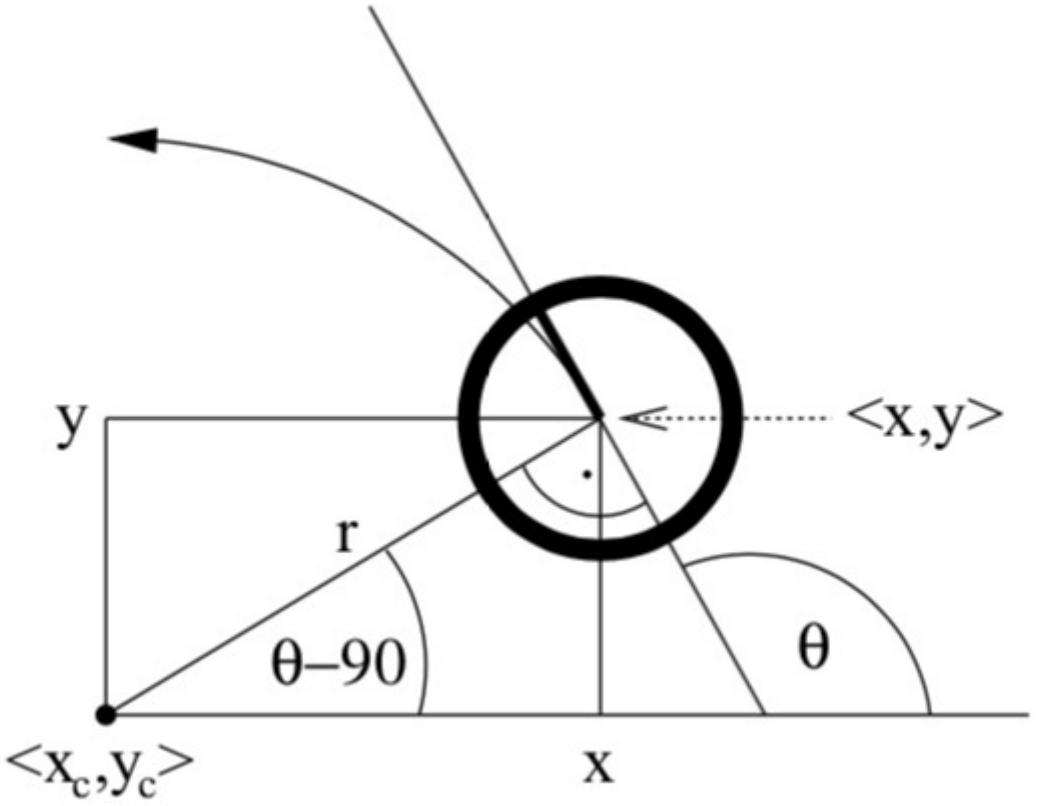
- init
- control
- point

```
nosetests -s unit.test_graph:test_graph_init
```

Input Data Interpret control event

Seminar.Task#1
Refer to: theory.pdf, Sec. “Car motion model”

r = v / w



Input Data Interpret init event

Seminar.Task#2
Refer to: theory.pdf, Sec. “Edge Fabrics”

```
class Constraint(object):
   def __init__(self, pose_edges=[], features=[]):
...

class SomeConstraintBuilder(IConstraintBuilder):
   def build(self):
        constraint = Constraint(pose_edges=[self._edge])
        self._edge = None
        self._ready = False
        return constraint
```

```
nosetests -s unit.test_edges:test_prior_edge
```

User Code Create Prior Edge

```
class Edge(object):
    def __init__(self, vertices):
        ....

    @property
    def inf(self):
        raise Exception('Not implemented')

    @property
    def error(self):
        raise Exception('Not implemented')

    def compute_error(self):
        raise Exception('Not implemented')
```

```
nosetests -s unit.test_edges:test_prior_edge
```

User Code Create Odometric Edge

Seminar.Task#3
Refer to: theory.pdf, Sec. “Inverse motion model of a car”

```
nosetests -s unit.test_edges:TestOdometry
```

User Code Create Odometric Edge

Seminar.Task#4
Refer to: theory.pdf, Sec. “Data”

```
class OdometryEdge(Edge):
    inf = None
    error = None

    def __init__(self, from_vertex, to_vertex, event):
```

```
nosetests -s unit.test_graph:
test_graph_optimization_without_landmarks
```

Homework

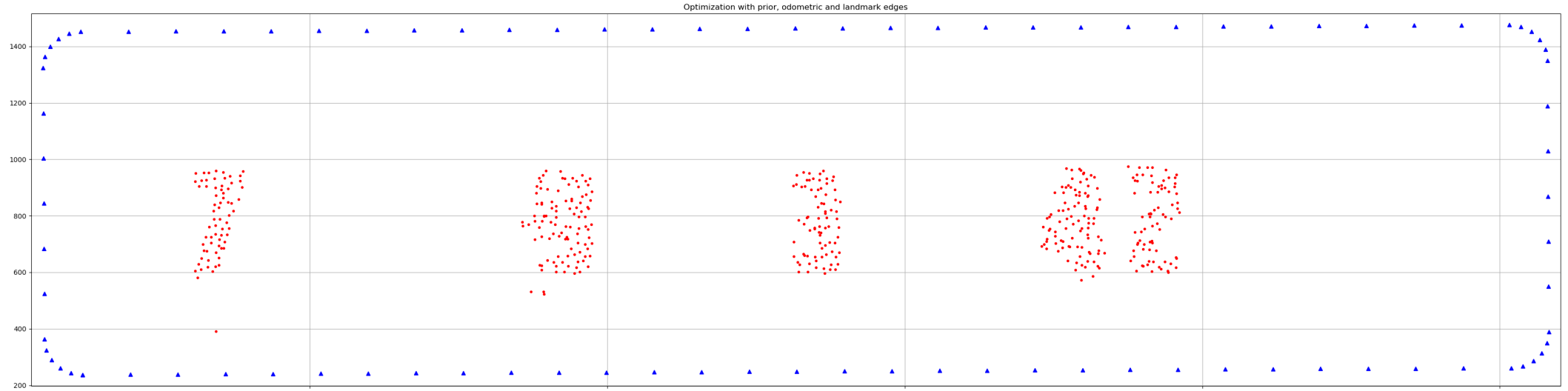
Grab Your Personal Task number here:

(Input your name in the last row, the row_id is a task number)

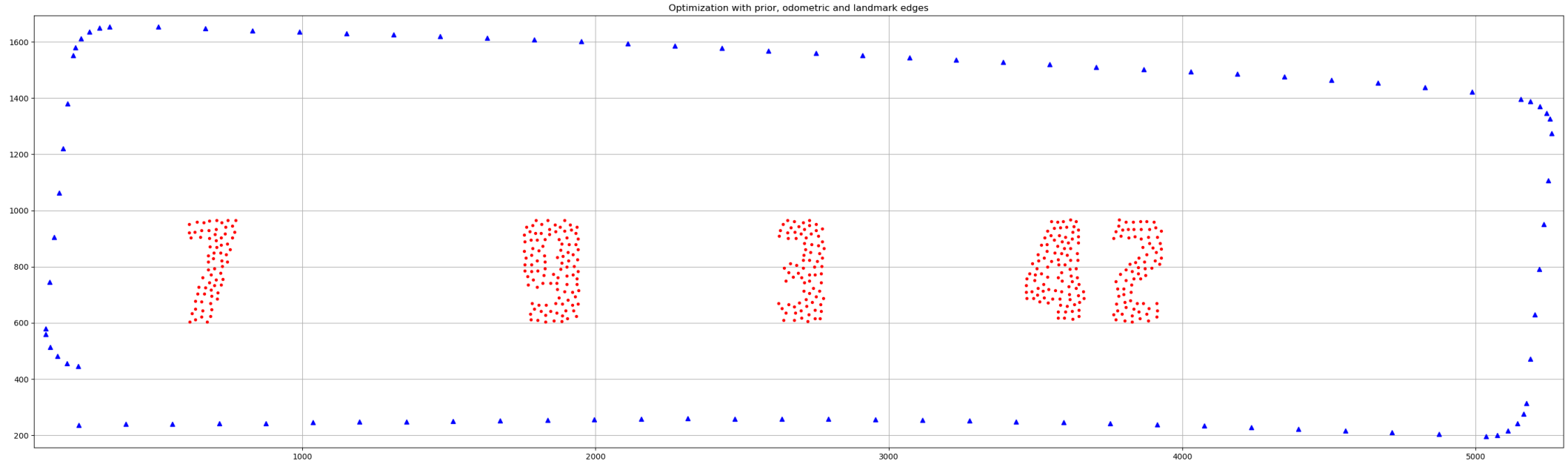
sdc_course_graph_slam_hw_Spring_2021.xlsx

https://tinyurl.com/ax59bzsv

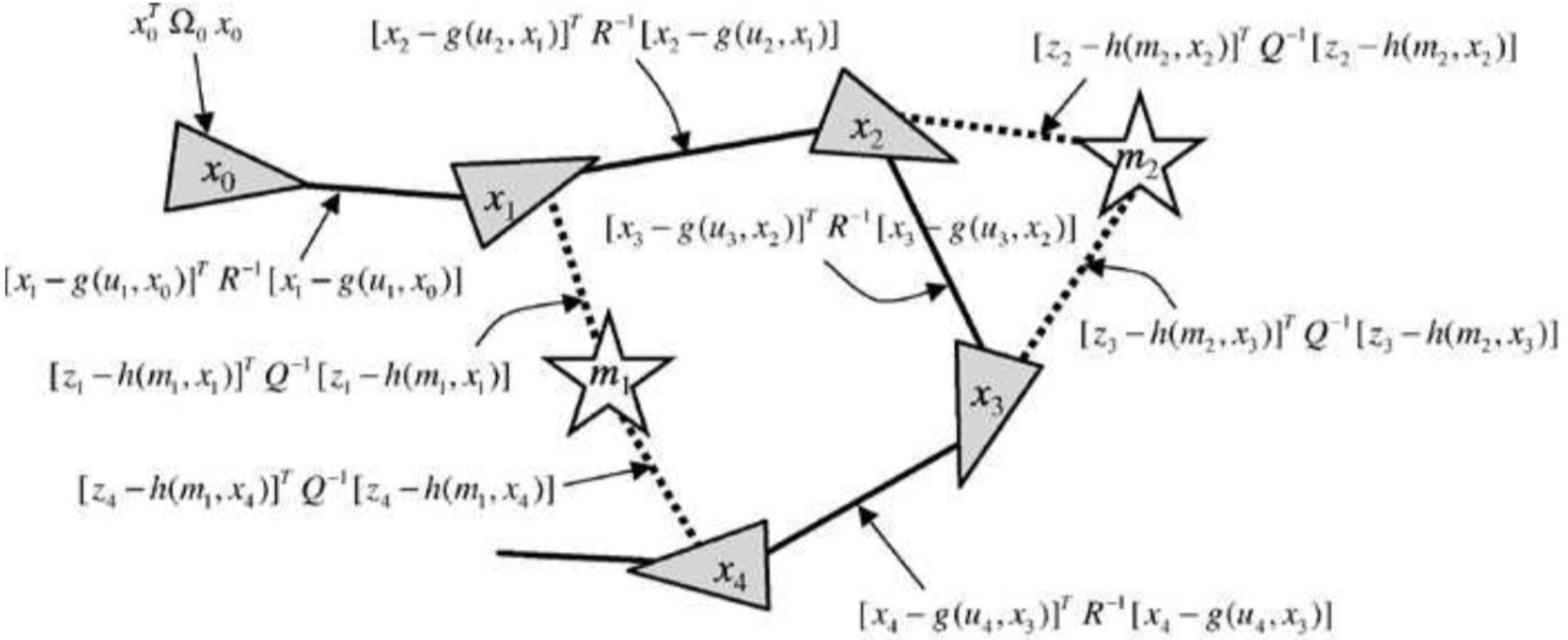
Homework What you'll improve on



To turn it into:



Optimization



Sum of all constraints:

$$J_{\text{GraphSLAM}} = x_0^T \Omega_0 x_0 + \sum_t [x_t - g(u_t, x_{t-1})]^T R^{-1} [x_t - g(u_t, x_{t-1})] + \sum_t [z_t - h(m_{c_t}, x_t)]^T Q^{-1} [z_t - h(m_{c_t}, x_t)]$$

Thank you

Dmitri Kovalenko
dmitri.a.kovalenko@gmail.com
tg: wfthirtyfour