Self Driving Car Course

Seminar on Graph SLAM

Dmitri Kovalenko

2022.03.23

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%
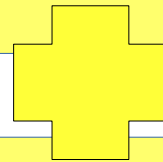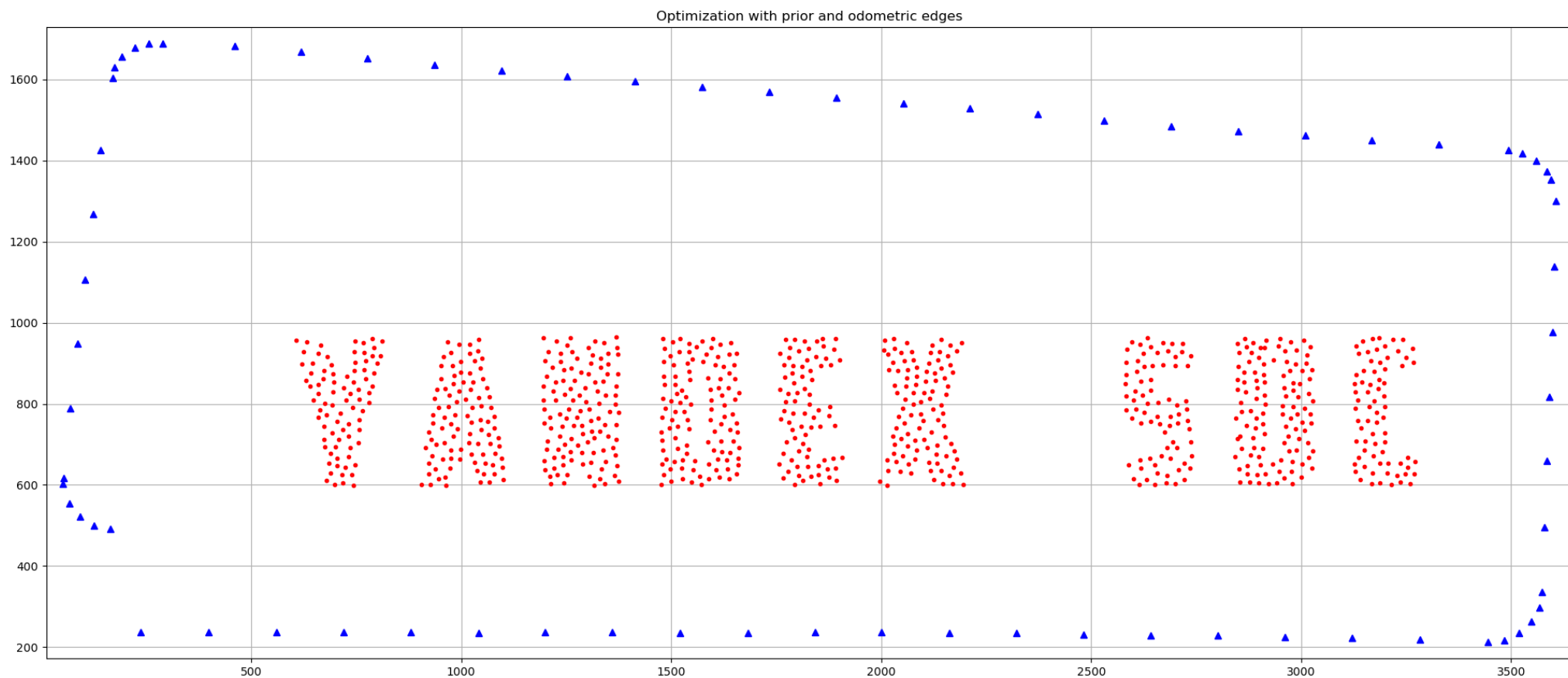
About the problem

Input Data	Computation	Result
timeline.json	Optimization	Trajectory & Map
	UserCode	

Result

Optimization with prior and odometric edges

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"
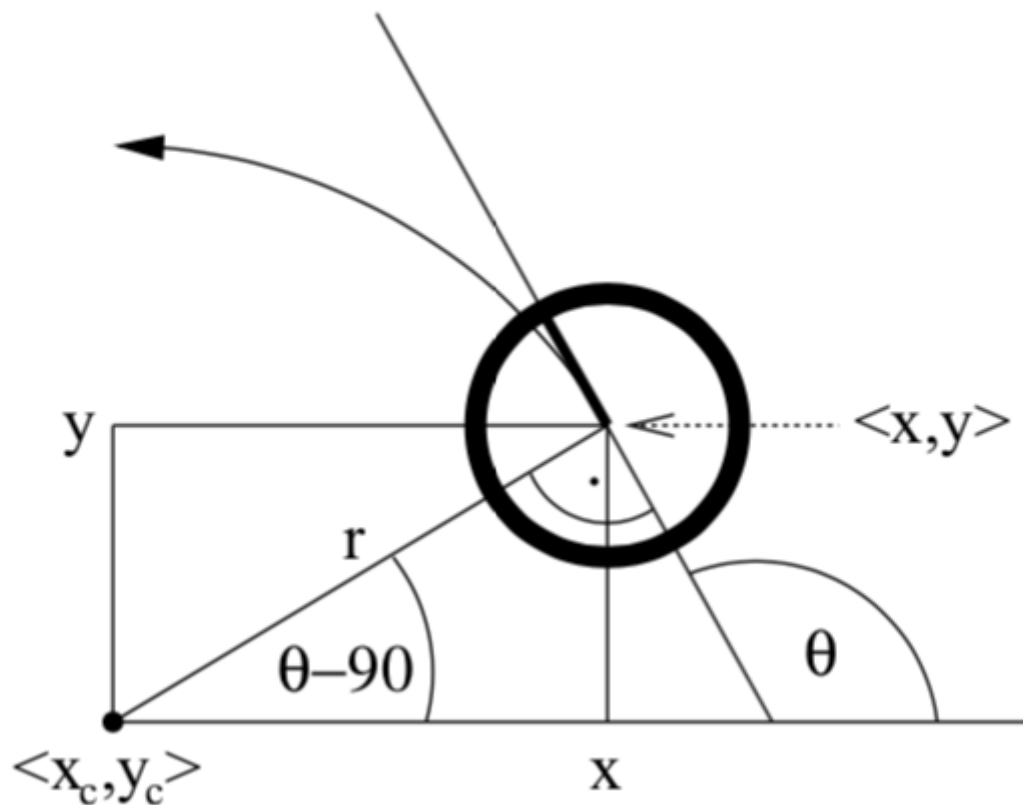
Event types:

- 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"

```python
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

```python
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"

```python
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:

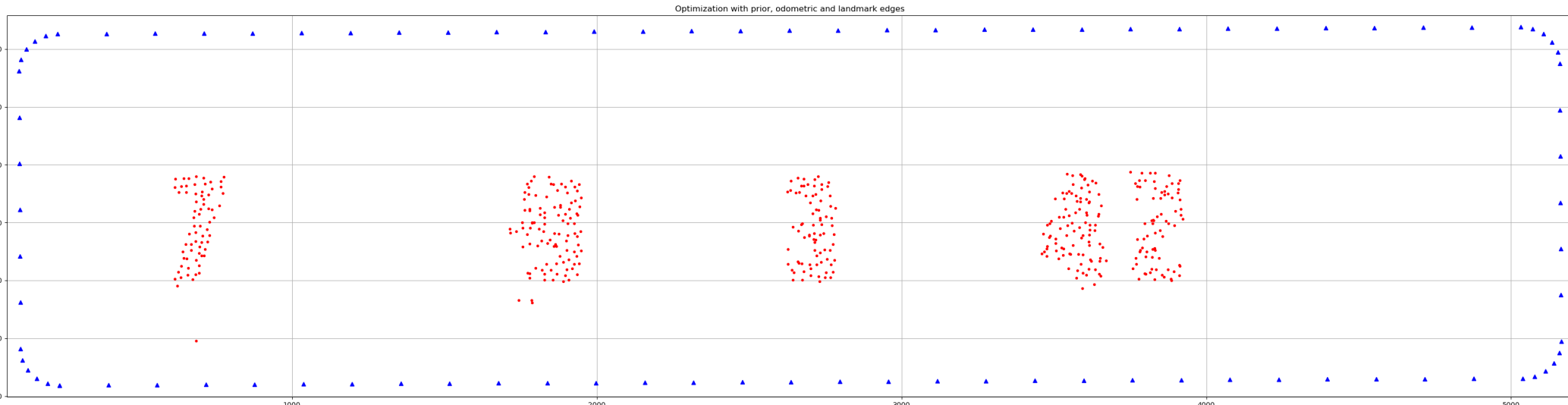- Input your name in the last row, row_id % 55 is the task number
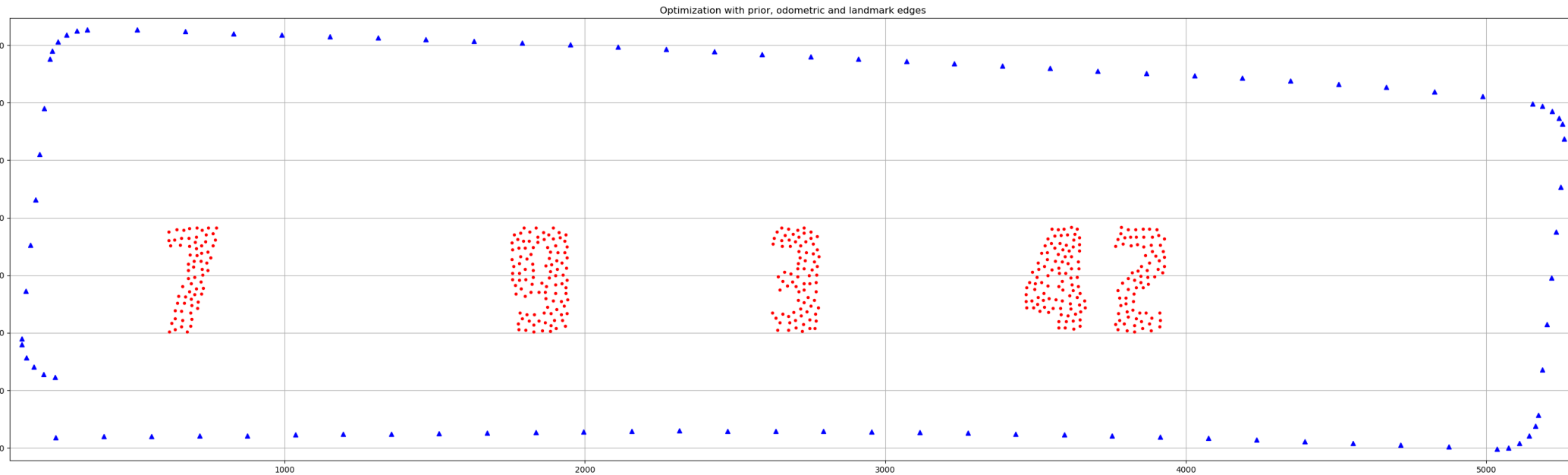
  sdc_course_graph_slam_hw_Spring_2022.xlsx

  https://tinyurl.com/4h8yb8jj

- Personal tasks are under: seminar02-graph-slam/code/data/timeline_{your task number}.json

Homework What you'll improve on

Optimization with prior, odometric and landmark edges

To turn it into:

Optimization with prior, odometric and landmark edges

Optimization
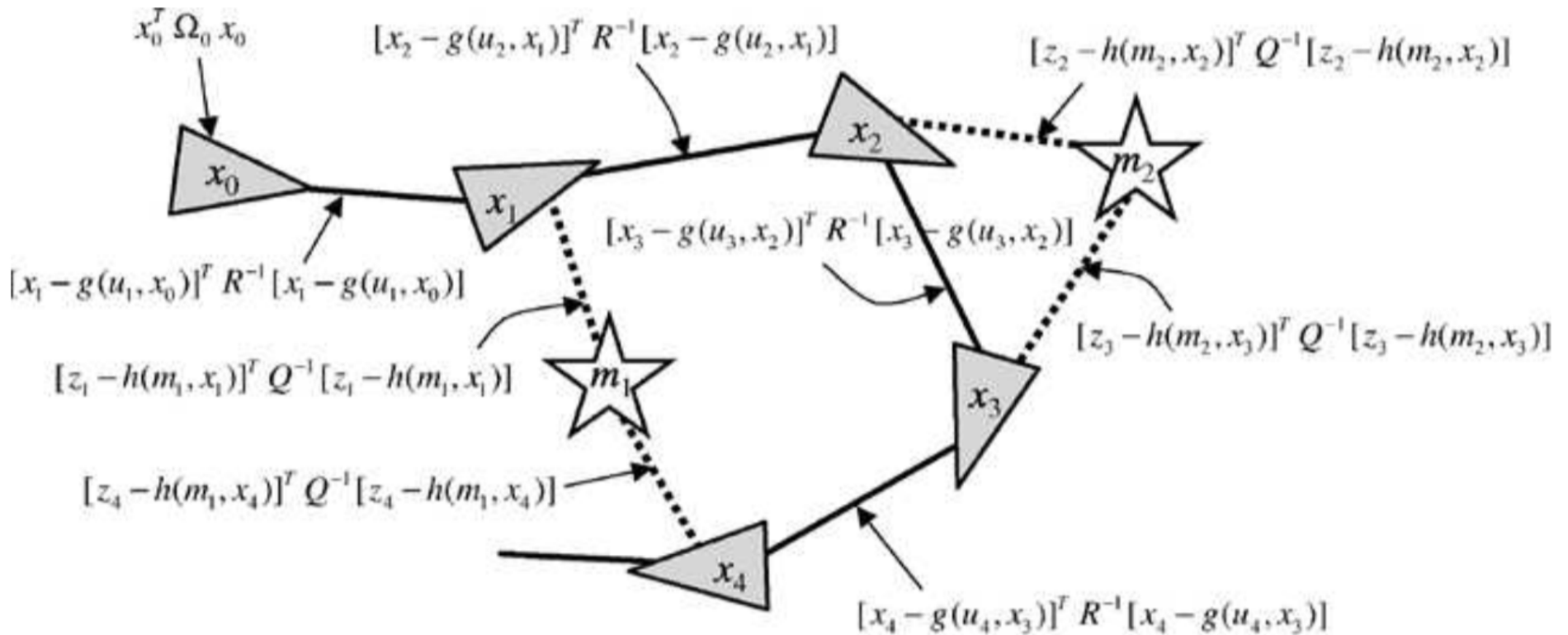
Sum of all constraints:

$$J_{\text{GraphSLAM}} = x_0^T \, \Omega_0 \, x_0 + \sum_i [x_i - g(u_i, x_{i-1})]^T \, R^{-1} [x_i - g(u_i, x_{i-1})] + \sum_i [z_i - h(m_{c_i}, x_i)]^T \, Q^{-1} [z_i - h(m_{c_i}, x_i)]$$

Thank you

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