Mining Constrained Regions of Interest: An optimization approach

Alexandre Dubray Guillaume Derval Siegfried Nijssen Pierre Schaus

Table of contents

Introduction

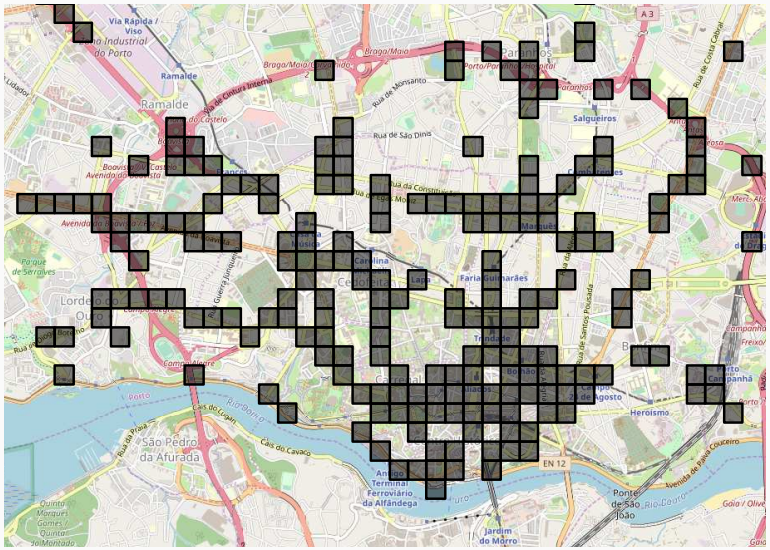
Motivations

- The amount of spatiotemporal data is exploding (smartphone applications, sport devices, fleet management, etc)
- There is a need to process more efficiently these data
- Rewrite the raw trajectories (GPS points) as sequence of *Regions of Interest* (ROI)

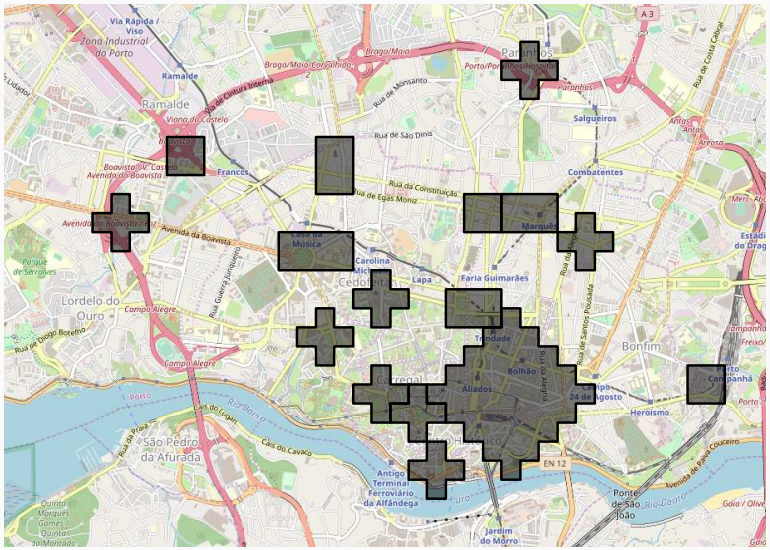
Preparation of the data

1. Divide the map with a grid
2. Assign a density value to each cell. A cell is dense if its density is above a threshold
3. Express the ROI as an aggregation of dense cells

Example of ROIs



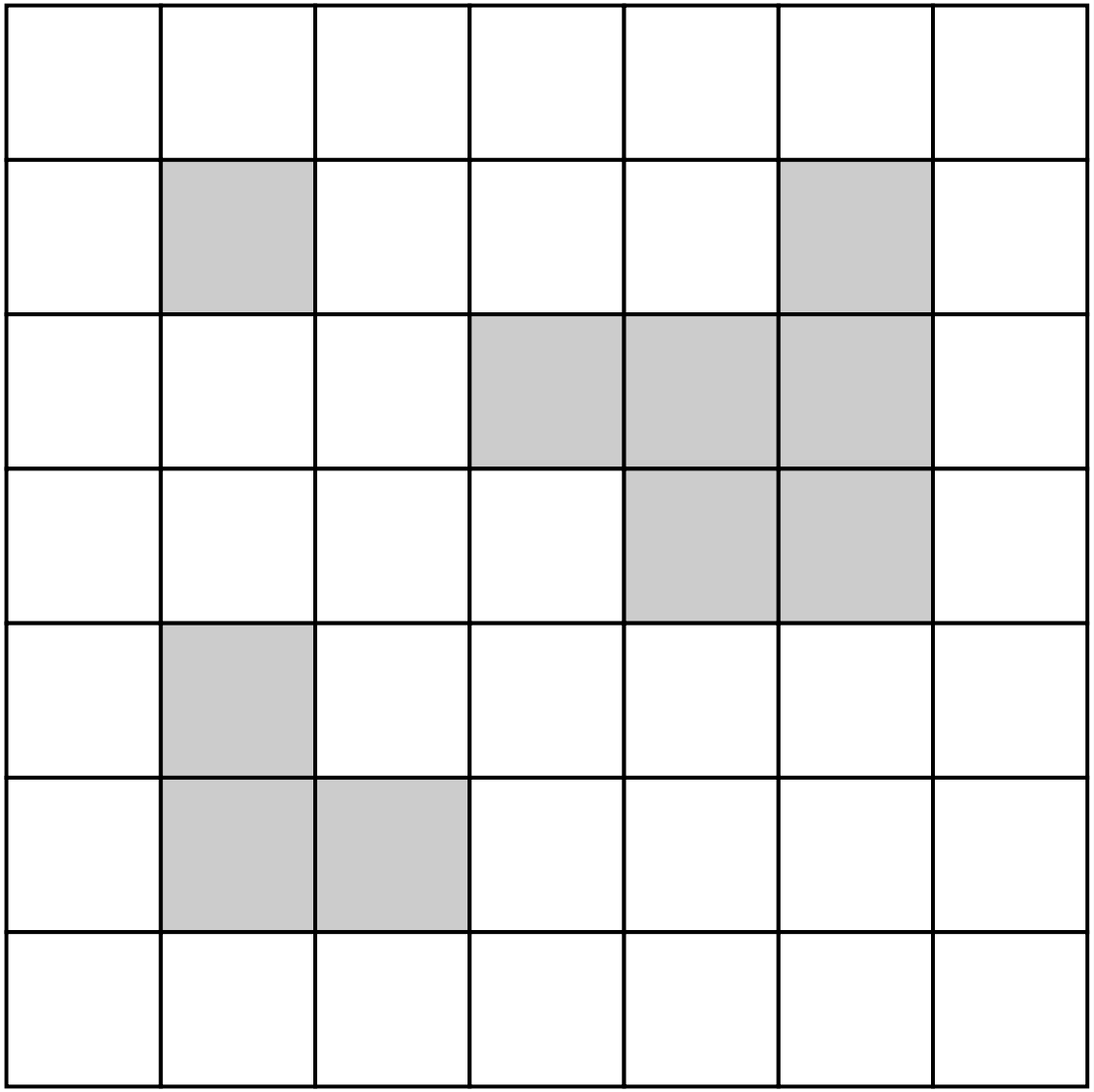
(a) Initial set of dense cells



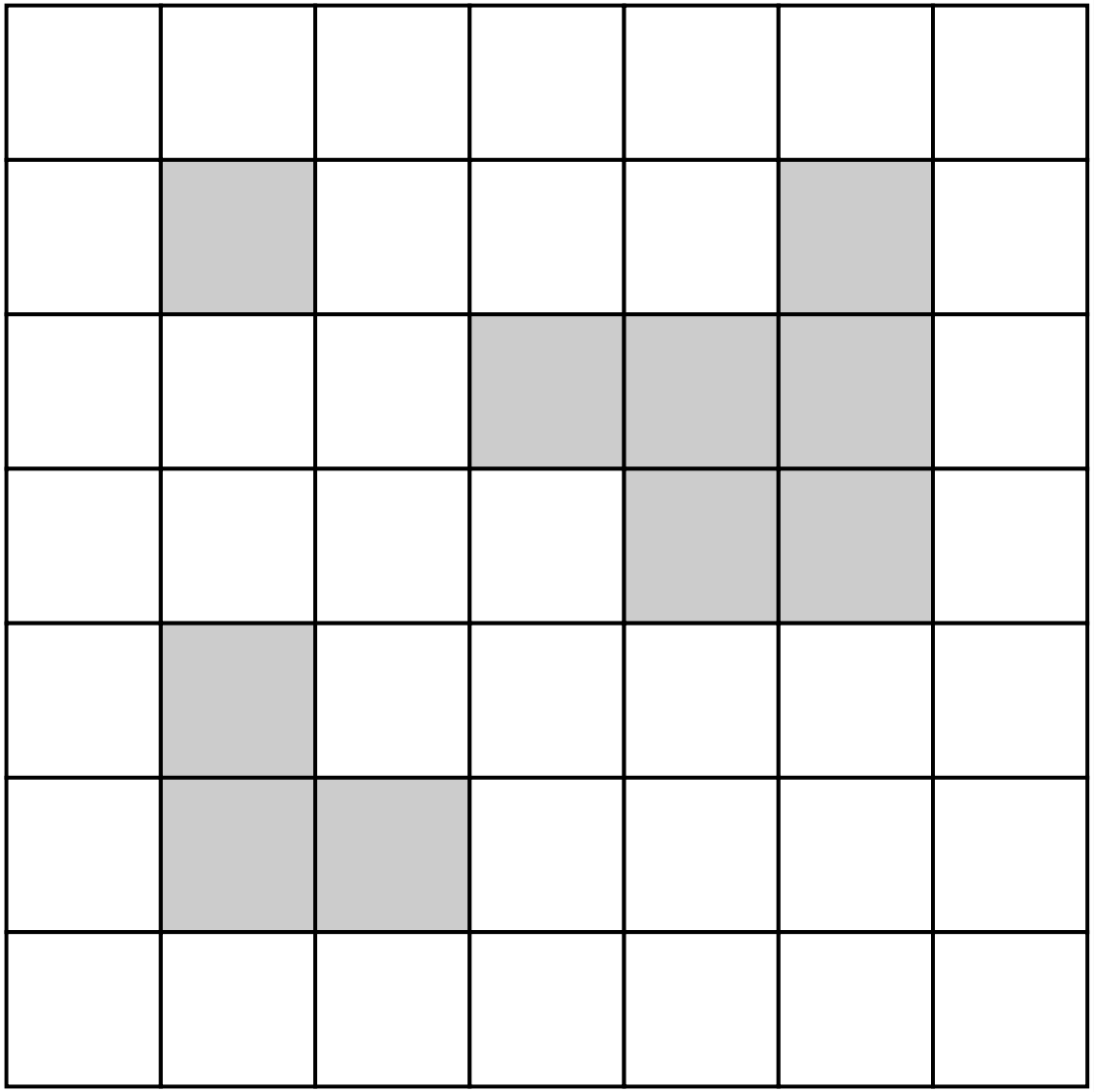
(b) Solution found by our method

PopularRegion

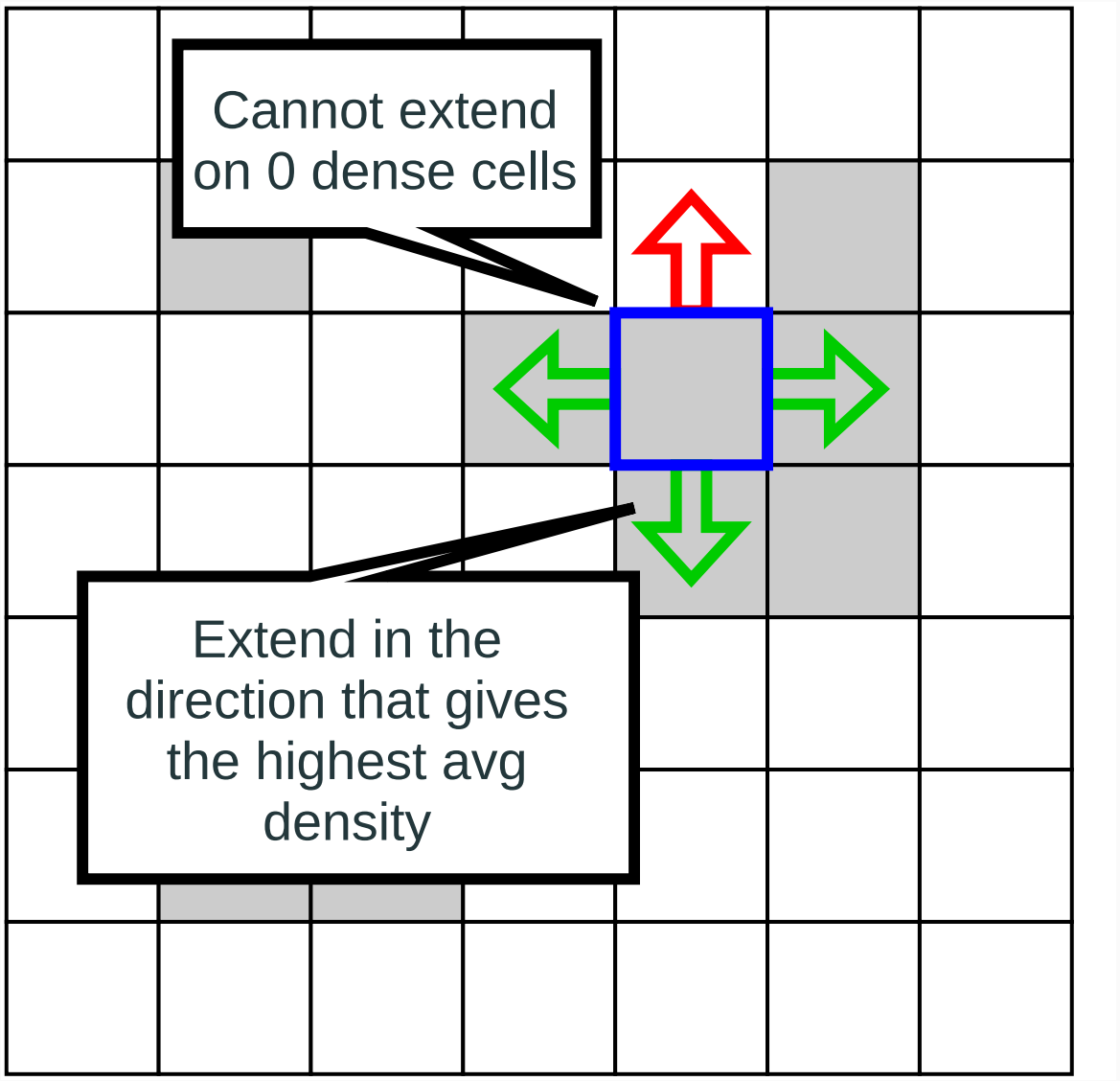
Execution of the algorithm



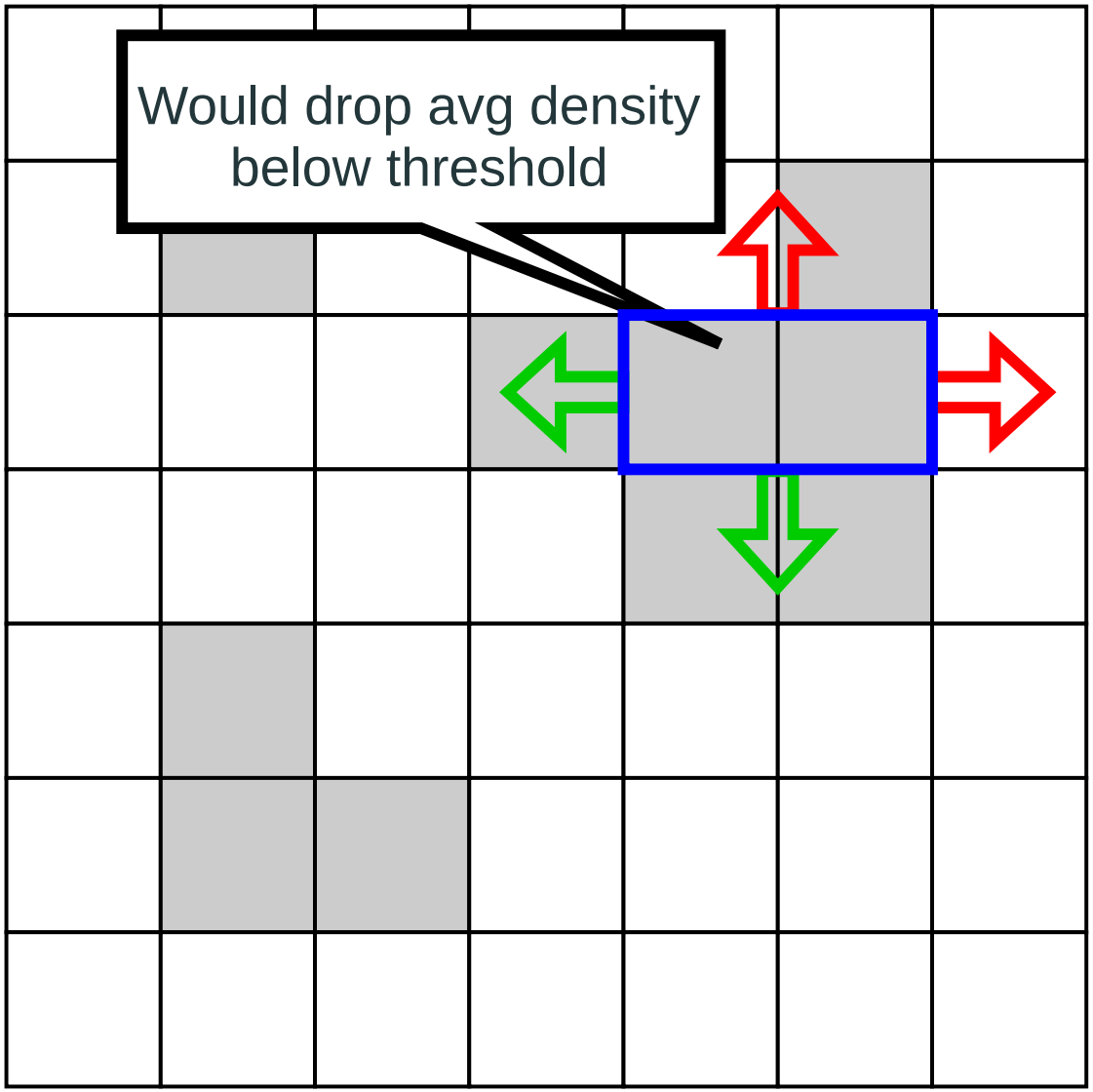
Execution of the algorithm



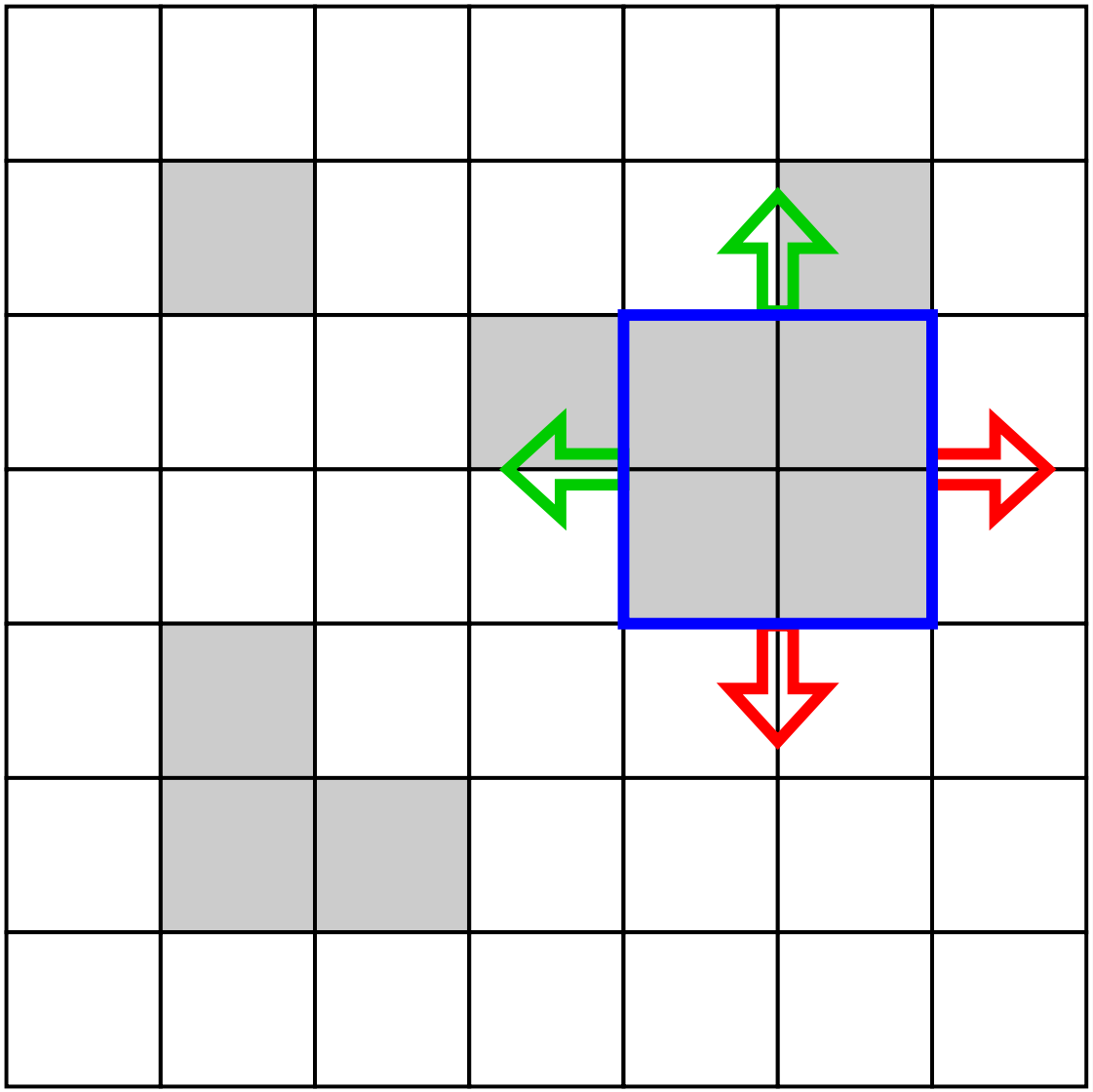
Execution of the algorithm



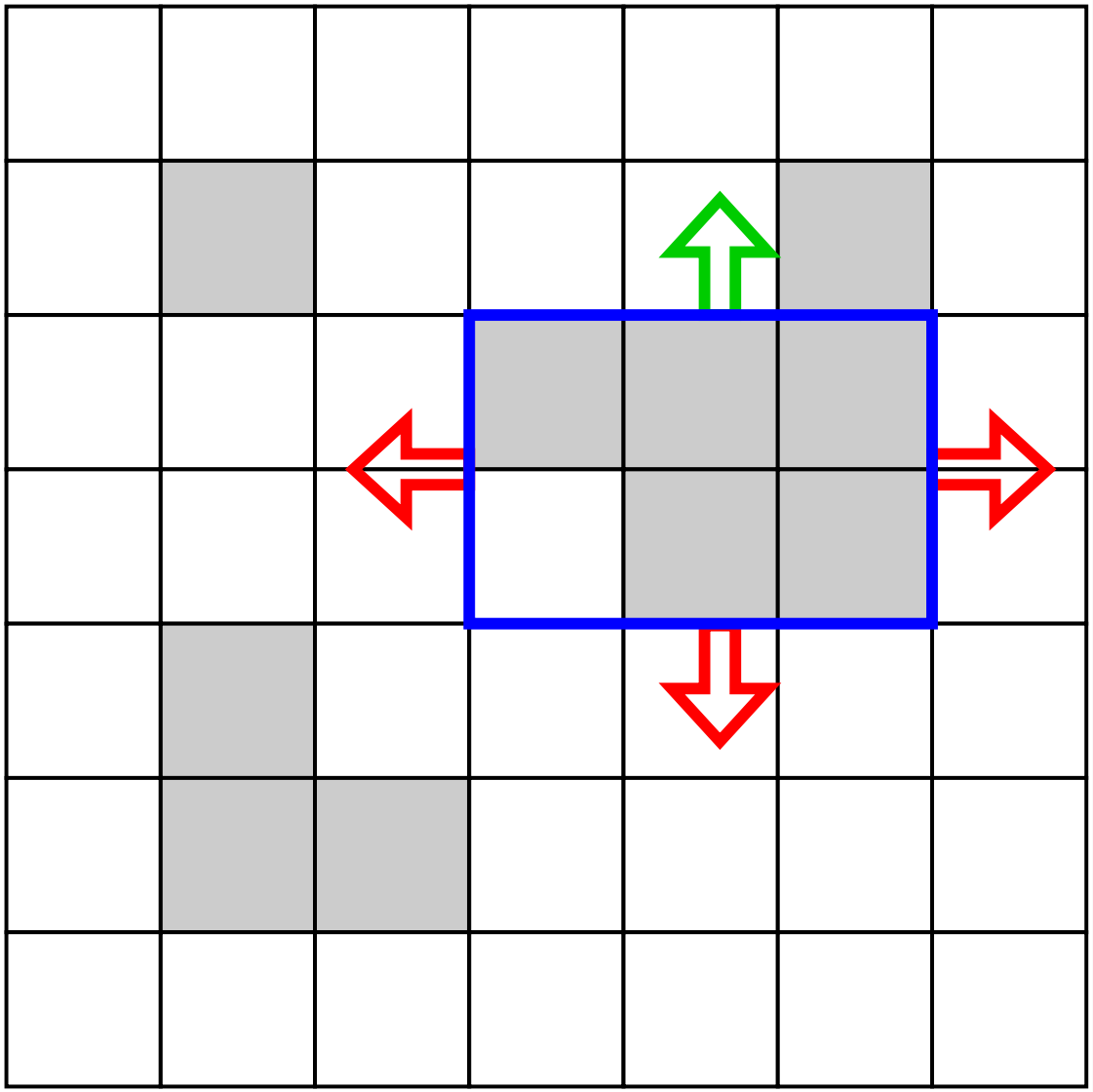
Execution of the algorithm



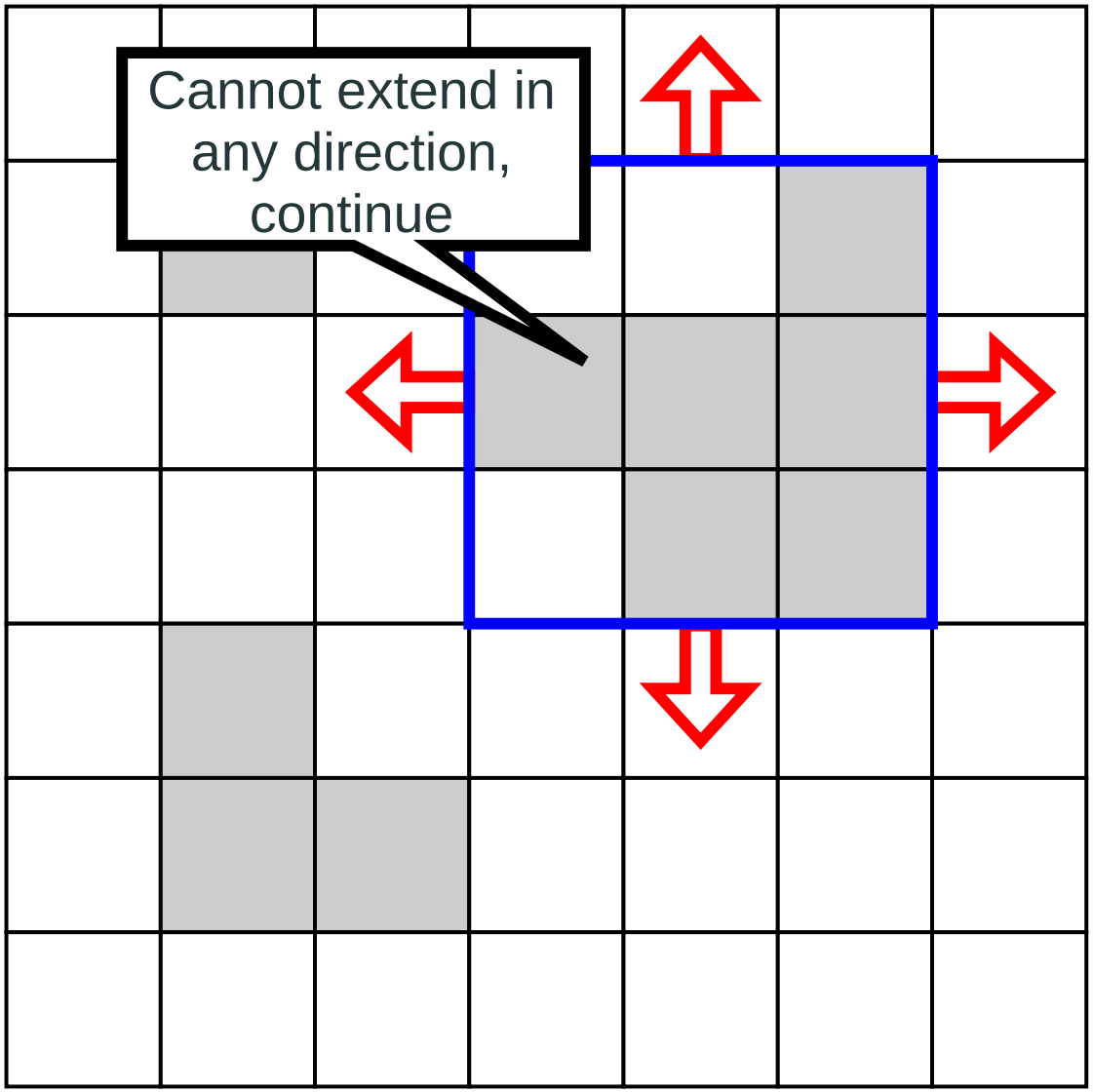
Execution of the algorithm



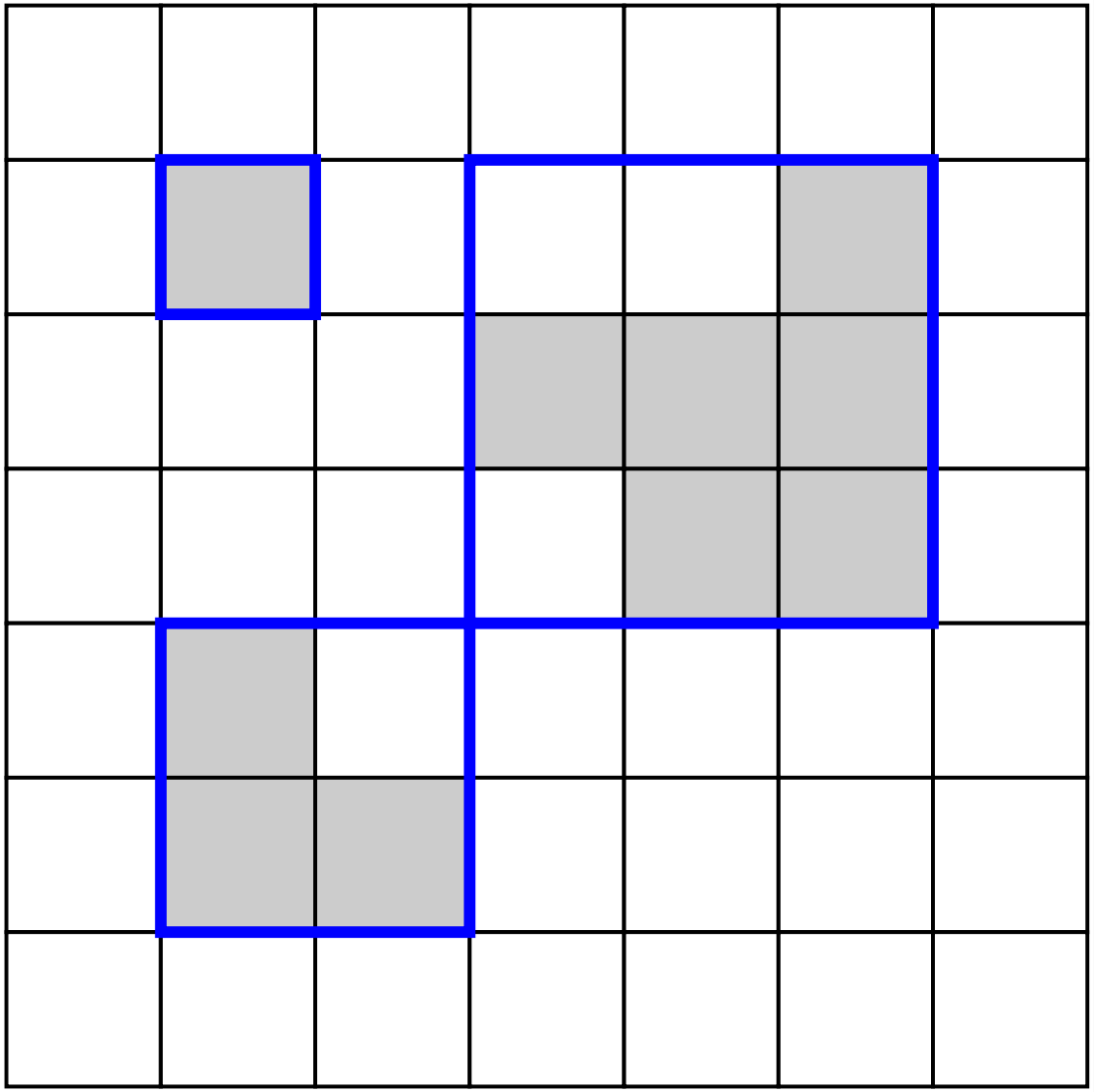
Execution of the algorithm



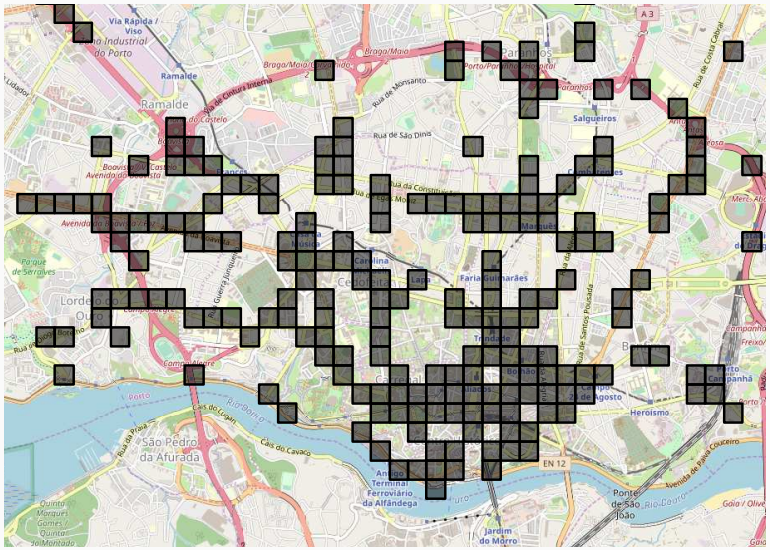
Execution of the algorithm



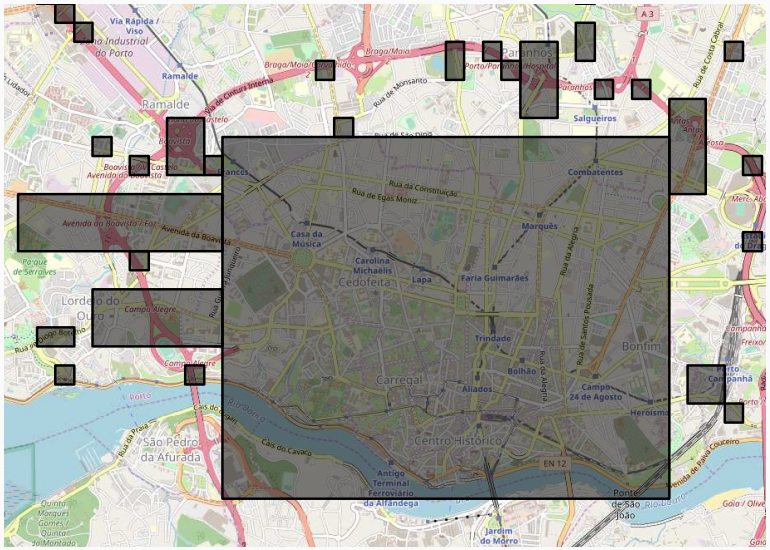
Execution of the algorithm



Result of the algorithm



(a) Initial set of dense cells



(b) Solution with 5% min average density

Advantages and disadvantages

- Scalable
- No formalization of the output
- Only rectangular regions
- Does not easily accept background knowledge

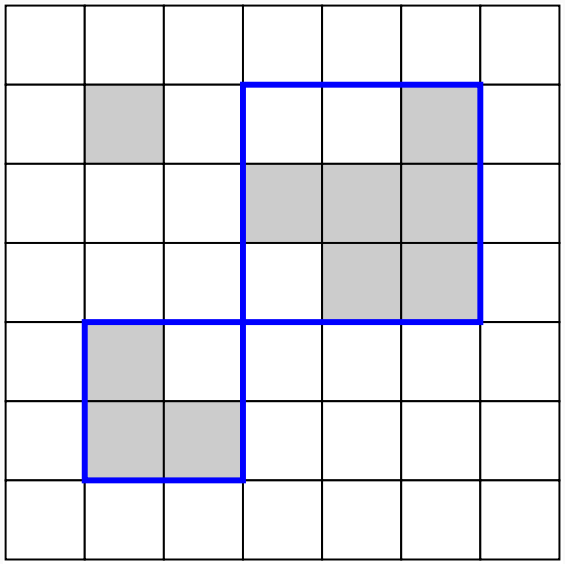
Our method

Outline

1. Generate a set of candidate ROI
    - Can have any shape
    - Impose *intra-ROI* constraints
2. Select from the candidates K final ROIs
    - Found by an optimization problem
    - Impose *inter-ROI* constraints

ROIs as an encoder

- The ROIs encode the dense status of the cells
- Example of encoding with two rectangles
  - 1 dense cells is not covered
  - 4 non-dense cells are covered
  - The encoding makes 5 errors
- We prefer encoding with less errors



Formalization of the problem (1)

Some notations:

- Let $\mathcal{G}$ be the grid, $\mathcal{S}$ a set of ROIs and θ the density threshold
- $error^{+} = \{c \in \mathcal{G} \mid density(c) \geq \theta \wedge c \notin \mathcal{S}\}$
- $error^{-} = \{c \in \mathcal{G} \mid density(c) < \theta \wedge c \in \mathcal{S}\}$

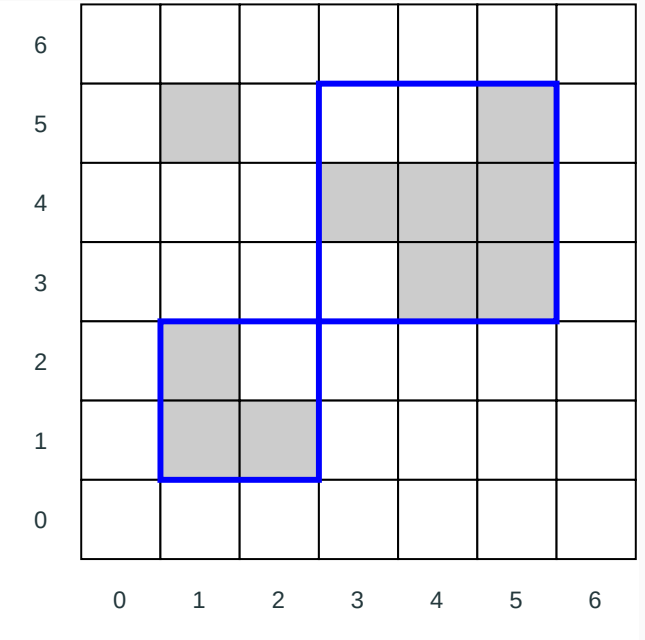
Formalization of the problem (2)

- Each cell $c \in \mathcal{G}$ is identified by 2 integers
- Length of the errors: $L(\mathcal{G} \mid \mathcal{S}) = 2 \cdot (|error^+| + |error^-|)$
- Length of the model: $L(\mathcal{S}) = \sum_{R_i \in \mathcal{S}} size(R_i)$
- Minimum Description Length principle tells that the best $\mathcal{S}$ is:

$$\arg\min_{\mathcal{S}} L(\mathcal{G} \mid \mathcal{S}) + L(\mathcal{S})$$

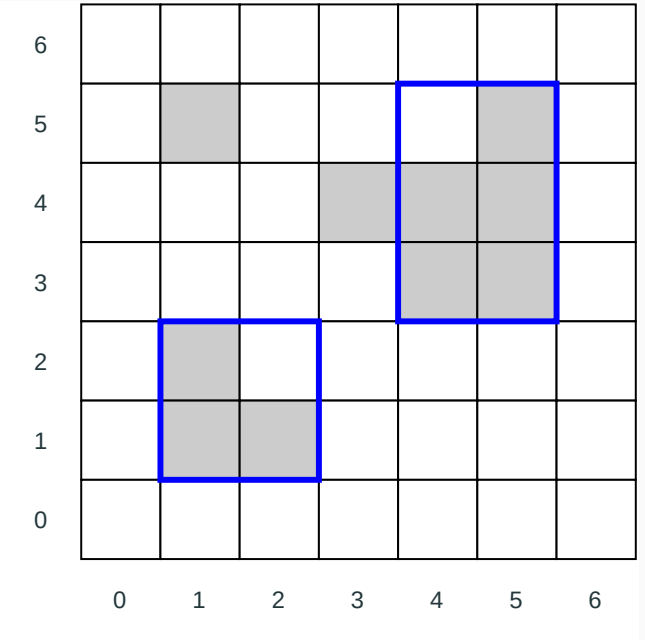
Example

- $\mathcal{S} = \{\langle(1,1),(2,2)\rangle, \langle(3,3),(5,5)\rangle\}$
- $L(\mathcal{S}) = 4 + 4 = 8$
- $L(\mathcal{G} \mid \mathcal{S}) = 2 \cdot (1 + 4) = 10$
- Total length of this model is $8 + 10 = 18$



A better model

- $\mathcal{S} = \{\langle(1,1),(2,2)\rangle, \langle(3,4),(5,5)\rangle\}$
- $L(\mathcal{S}) = 4 + 4 = 8$
- $L(\mathcal{G} \mid \mathcal{S}) = 2 \cdot (2 + 2) = 8$
- Total length of this model is $8 + 8 = 16$



Generation of the candidates

- The determinent factor for a candidate R_i is its contribution to the description length
- We can use any shape as long as we can compute this value
- In the generation of the candidates, we apply *intra-ROI* constraints

The optimization model

- 1 binary decision variable x_i per candidate R_i
- d_i = number of dense cells covered by candidate R_i
- u_i = number of non-dense cells covered by candidate R_i
- $size(R_i)$ = number of integer to encode R_i

$$\text{minimize} \sum_{R_i \in \mathcal{S}} x_i \cdot (2(u_i - d_i) + size(R_i))$$

subject to

$$\sum_{R_i \in \mathcal{S} | c \in R_i} x_i \leq 1 \qquad \forall c \in \mathcal{G}$$

$$x_i \in \{0, 1\} \qquad \forall R_i \in \mathcal{S}$$

Comparions

Setup

- Two versions of our method
  - With only rectangular regions
  - With rectangular and circular regions
- Showing results on Kaggle taxis dataset ($\approx$1.6 million trajectories)
- Comparing with PopularRegion and OPTICS (when clustering the dense cells)

Execution time

Minimum density threshold	2%			5%		
Grid side size	100	150	200	100	150	200
Number of dense cells ($\|\mathcal{G}^*\|$)	571	597	537	230	178	137
Number of ILP candidates	23 814	7 779	3 399	2 880	1 232	434
ILP optimization time (s)	4.328	0.464	0.109	0.113	0.044	0.029
PopularRegion run time (s)	0.003	0.005	0.006	0.002	0.003	0.004
OPTICS run time (s)	0.209	0.222	0.200	0.084	0.065	0.051

Description Length

- For high density threshold, number of errors becomes similar
- ILP-based methods produce smaller models
- Overall the Description Length is inferior for ILP-based methods

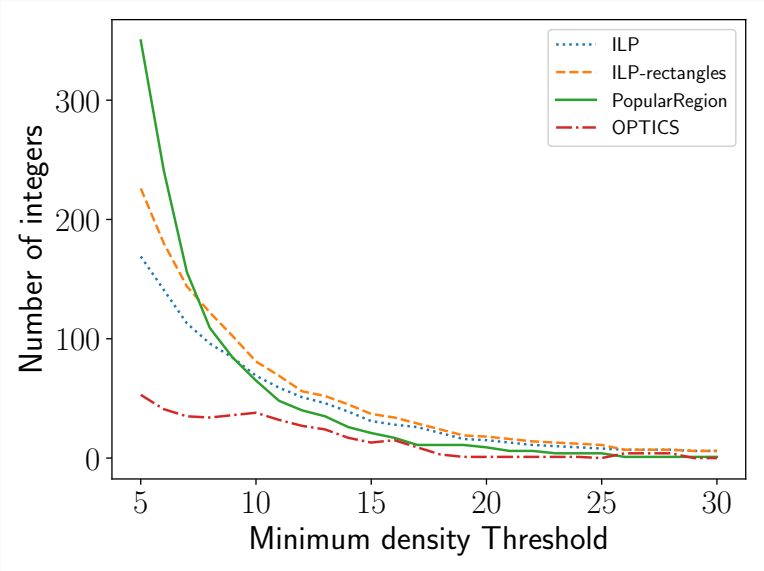


Figure 3: Encoding of the errors

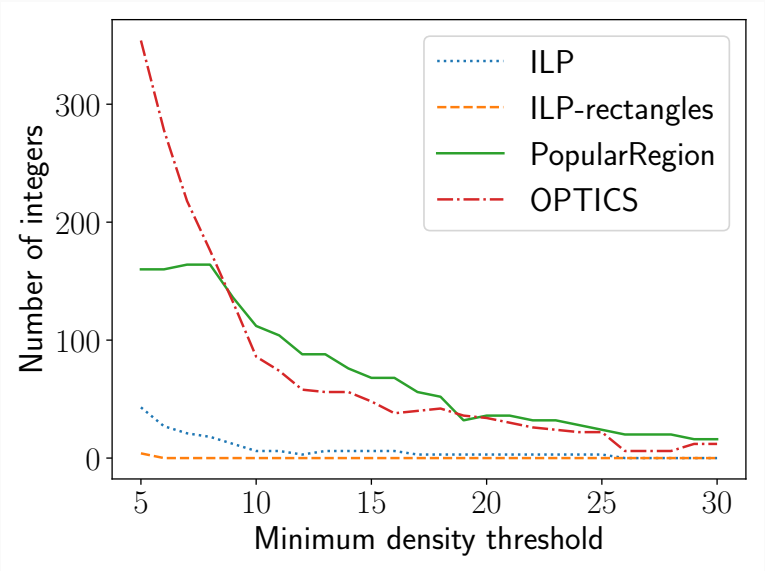


Figure 4: Encoding fo the models

Robustness to noise

- Start from a 100×100 grid
- Move every element of the trajectories to a neighboring cell with a probability p
- Choose the new cell randomly in a square of size 10 around the initial cell
- Recompute solution and compare to initial solution (with min density threshold 5%)

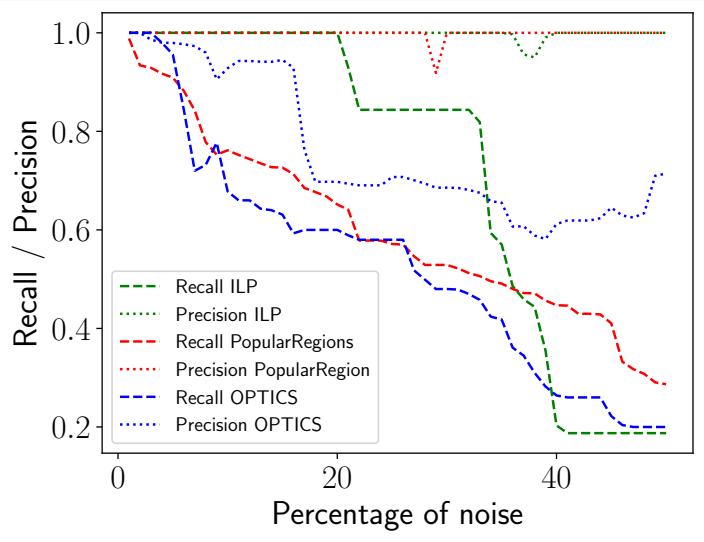


Figure 5: Recall and precision

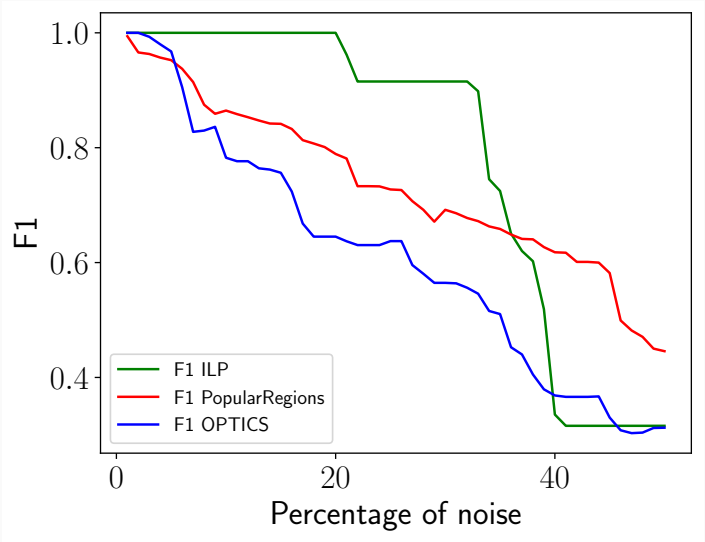


Figure 6: F1-measure