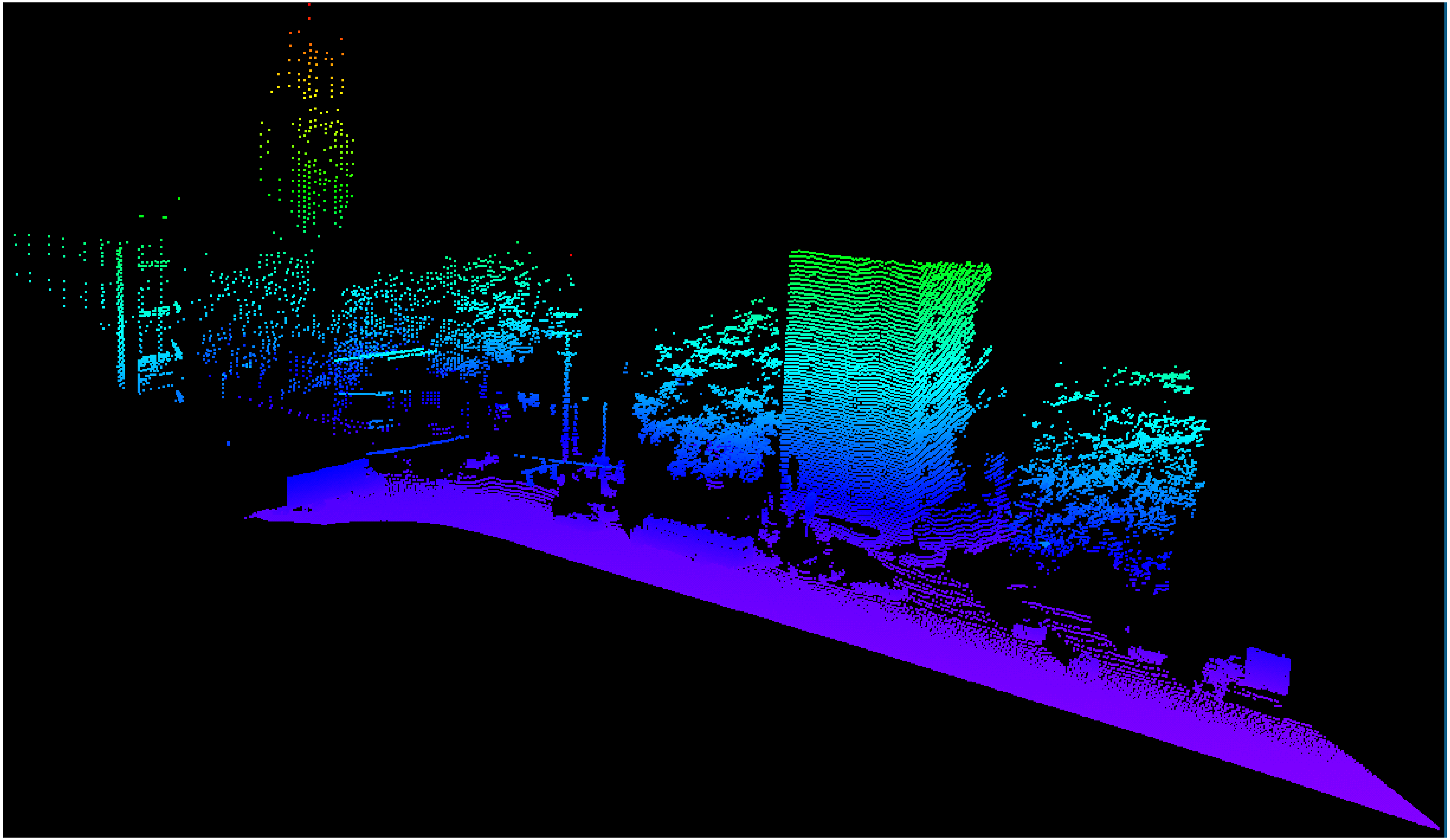
Onboard Contextual Classification of 3-D Point Clouds with Learned High-order Markov Random Fields

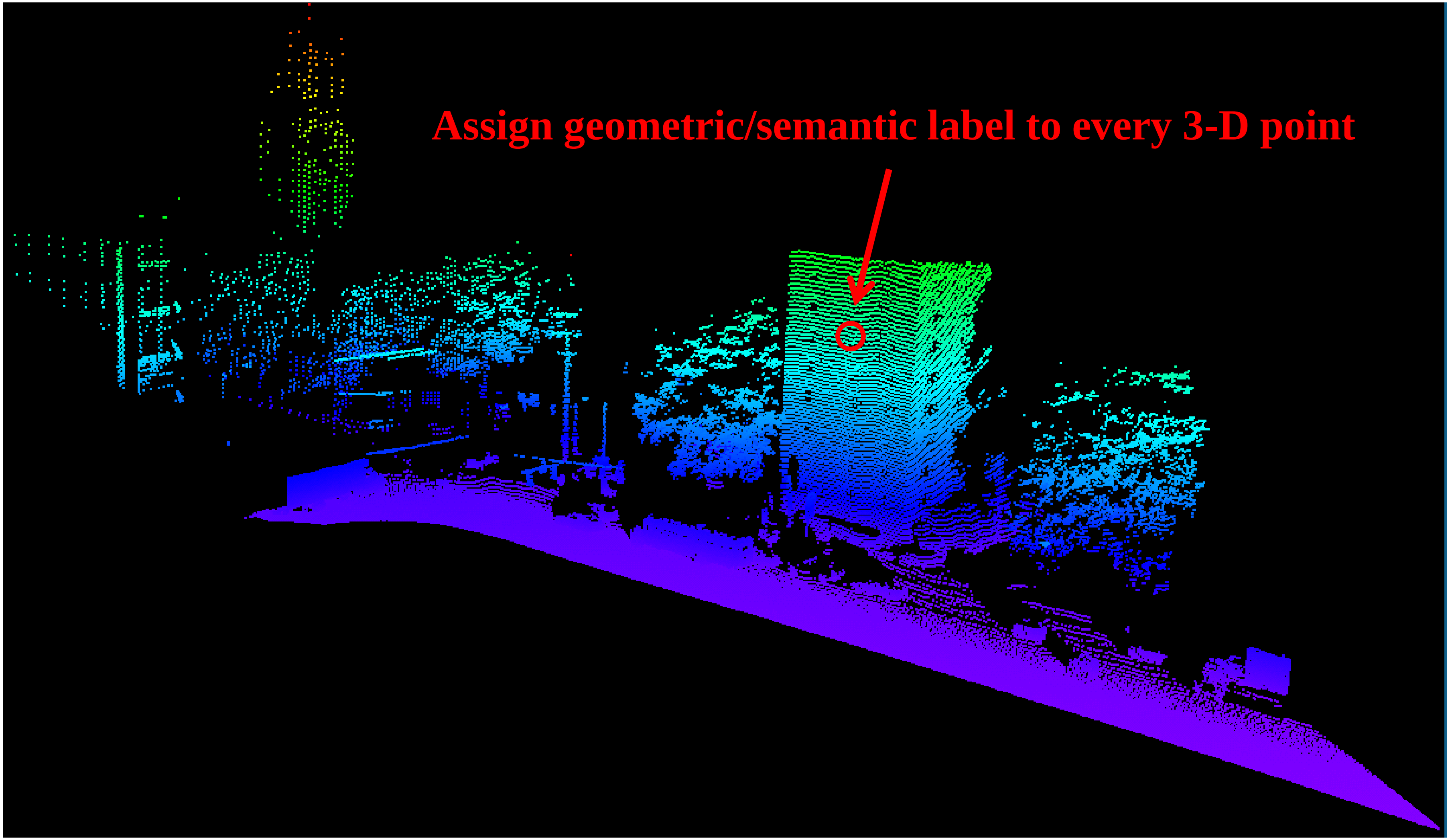
Daniel Munoz **Nicolas Vandapel** **Martial Hebert**

Carnegie Mellon
THE ROBOTICS INSTITUTE

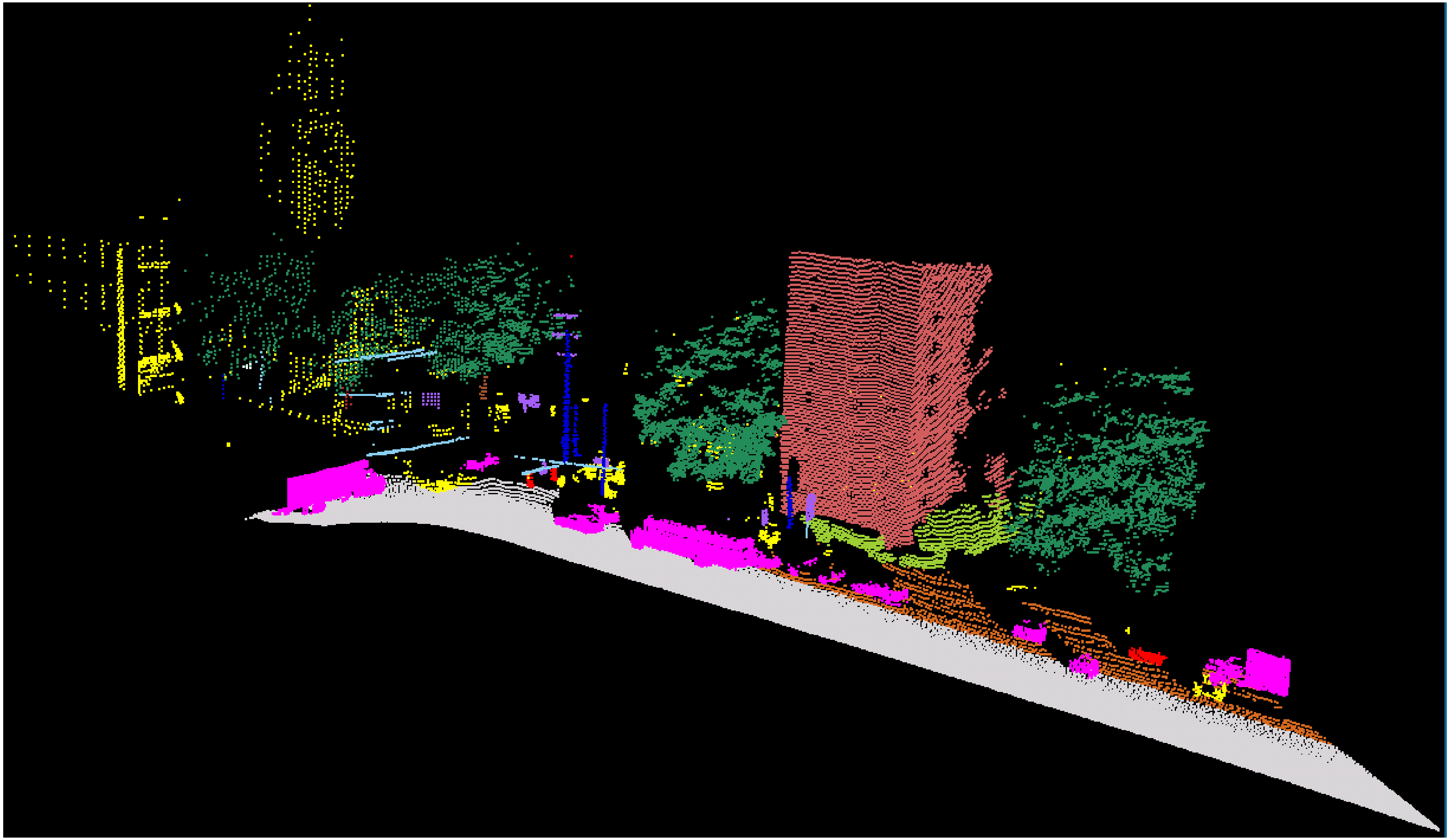
Example of 3-D point cloud



Problem: Automated 3-D point labeling

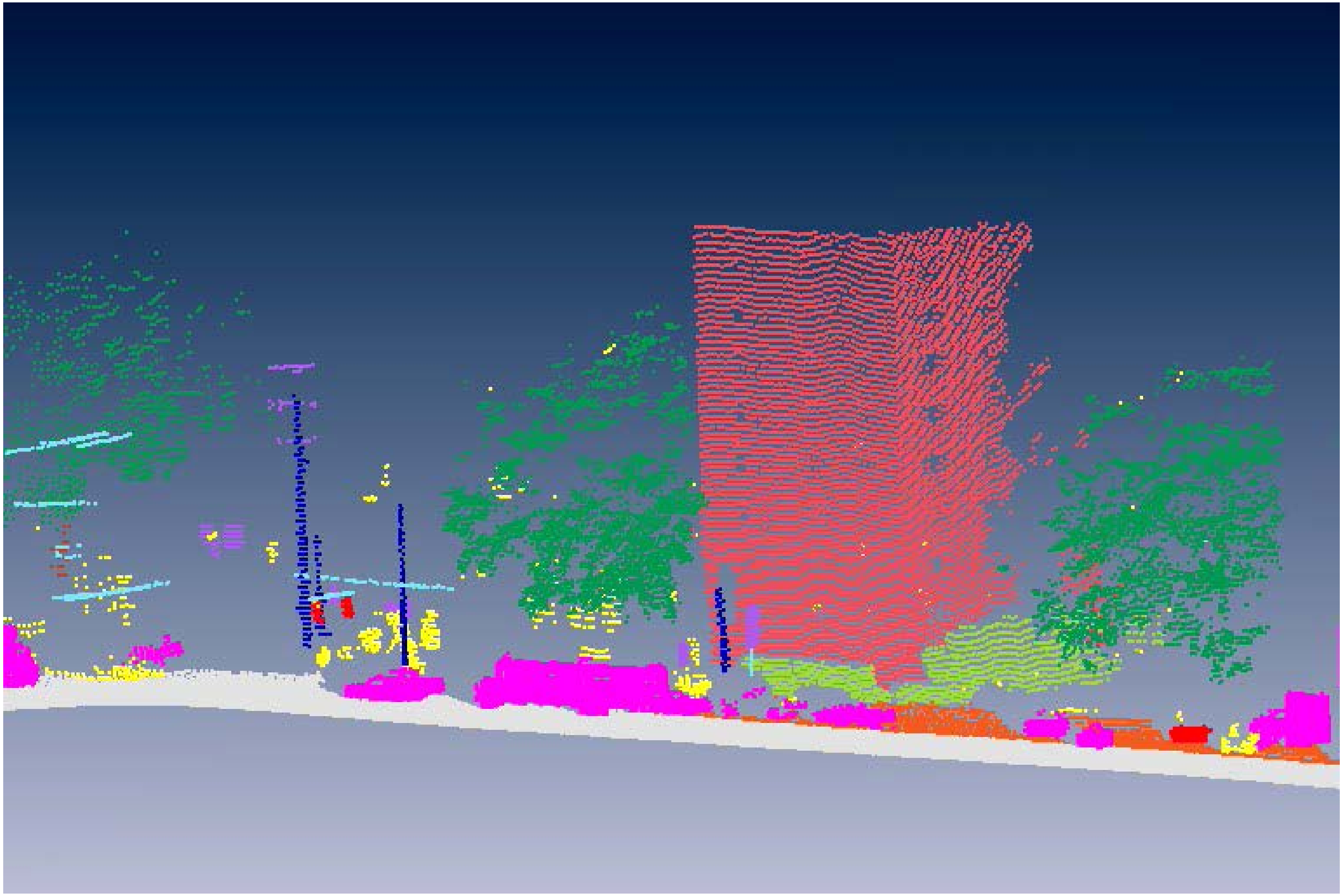


Problem: Automated 3-D point labeling



Hand labeled data

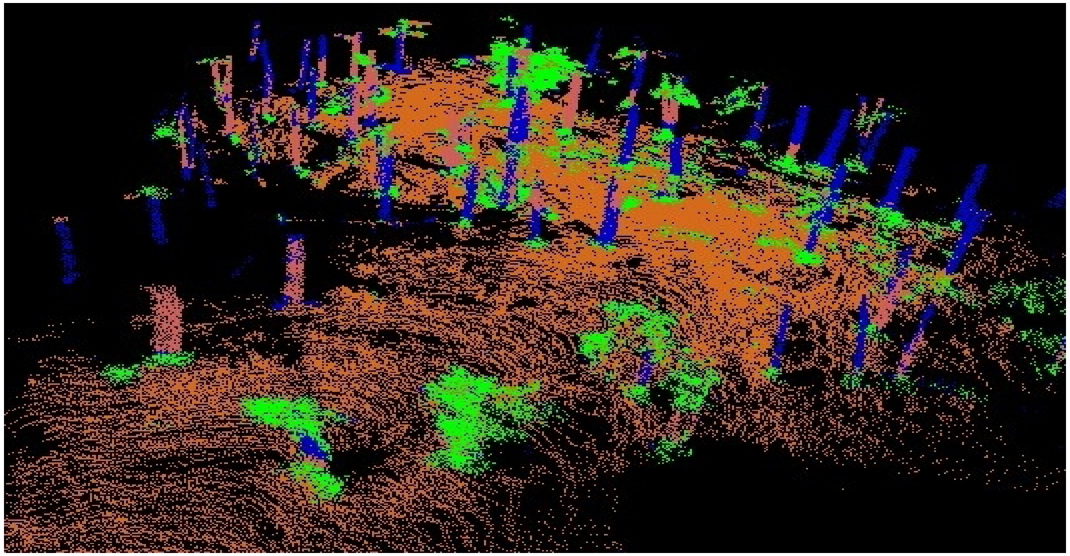
Problem: Automated 3-D point labeling



Hand labeled data

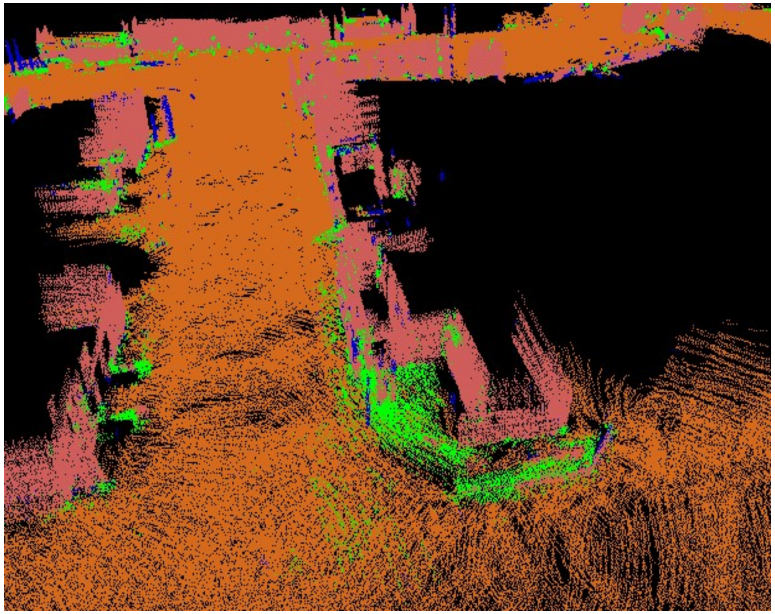
Problem: Automated 3-D point labeling

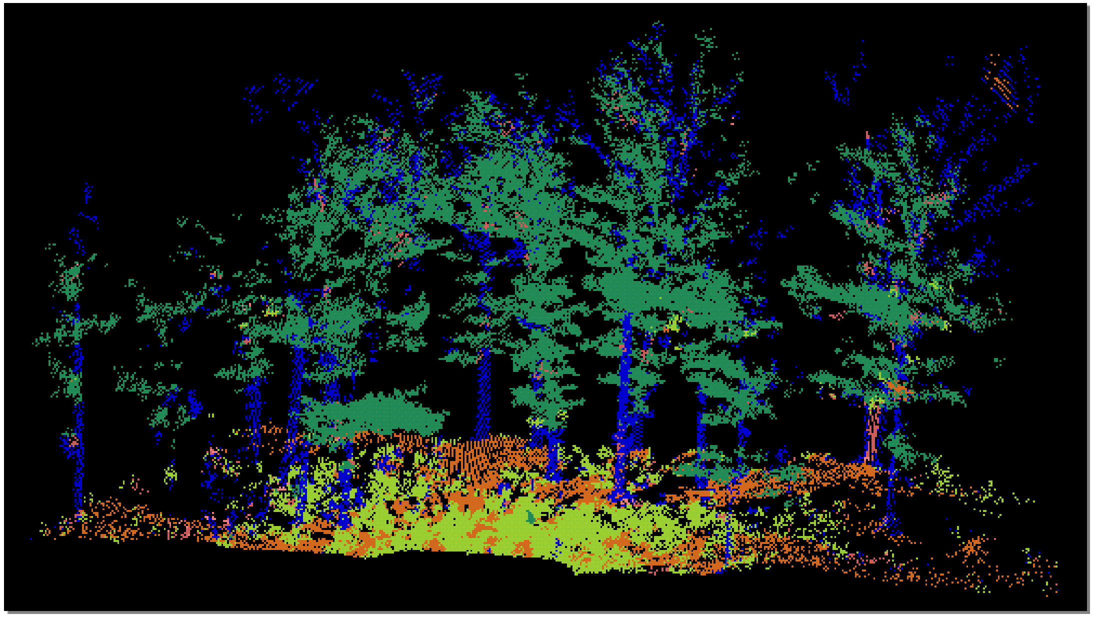
- Do it **onboard**



Scene understanding for autonomous vehicle navigation

- **Environments**: urban and natural settings





- **Labels**

Grass	Wire	Pole/Trunk	Ground	Shrub
Foliage	Facade	Wall	Roof	

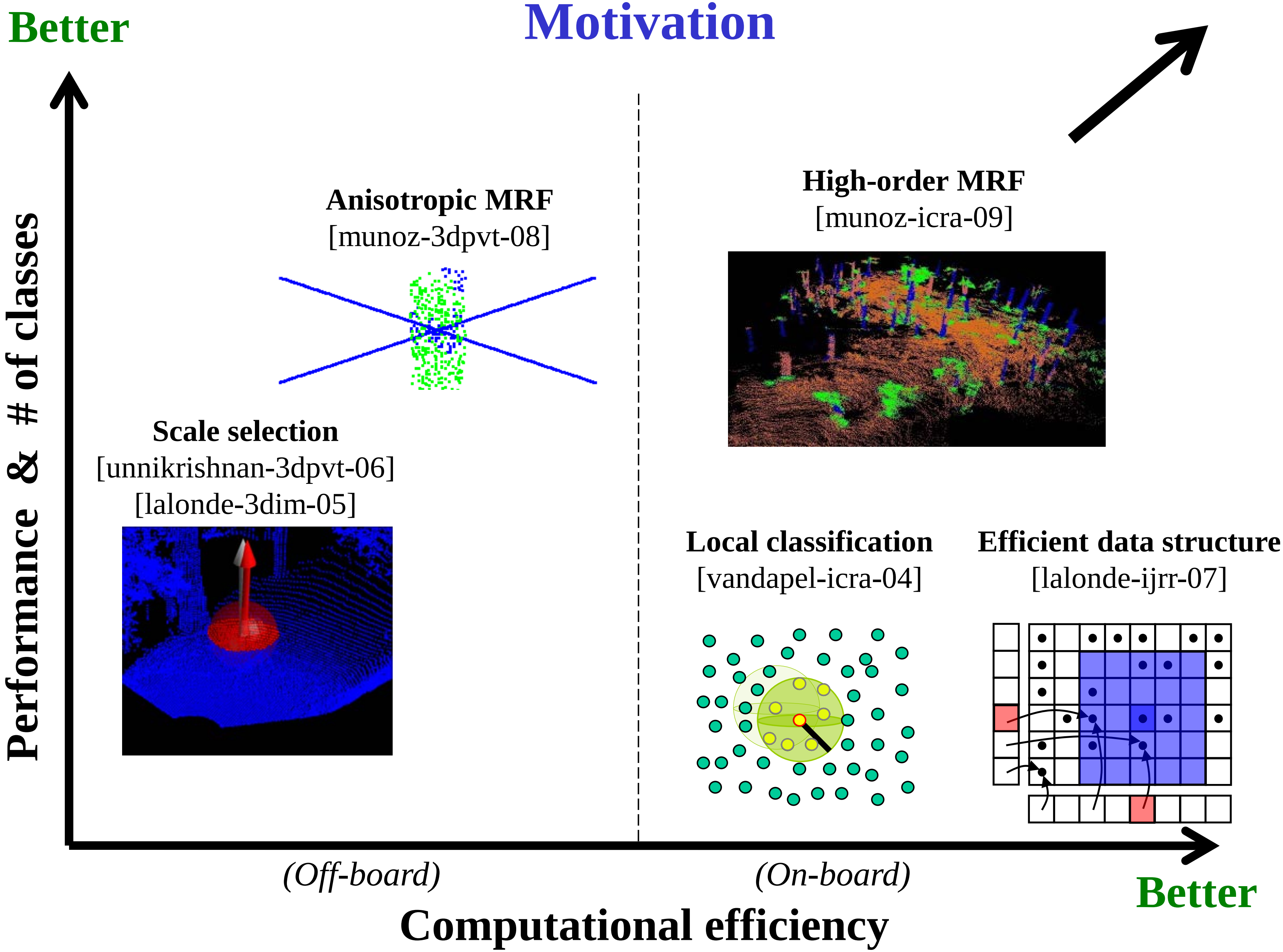
- **Purpose**: environment modeling, obstacle detection

Challenges

- Mobility laser data only
- Onboard data processing
  - Process continuously streaming data, over 100 K pts/s
  - Real-time data processing
  - Vehicle speed up to 6 m/s (20 km/h)

Demo-III eXperimental Unmanned Vehicle (Demo-III XUV)

Motivation



Better
Performance & # of classes
Anisotropic MRF
[munoz-3dpvt-08]
High-order MRF
[munoz-icra-09]
Scale selection
[unnikrishnan-3dpvt-06]
[lalonde-3dim-05]
Local classification
[vandapel-icra-04]
Efficient data structure
[lalonde-ijrr-07]
(Off-board)
(On-board)
Better
Computational efficiency

Outline

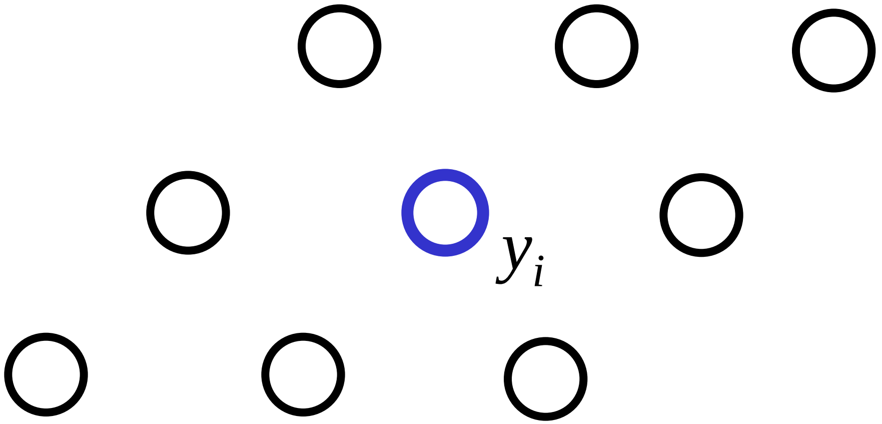
- Model introduction
- Contributions
- Onboard experiments

Model introduction

- Local classifiers

$$\mathbf{Y} = \{Y_1, \ldots, Y_N\}$$

$$Y_i \in \{\ell_1, \ldots, \ell_K\}$$



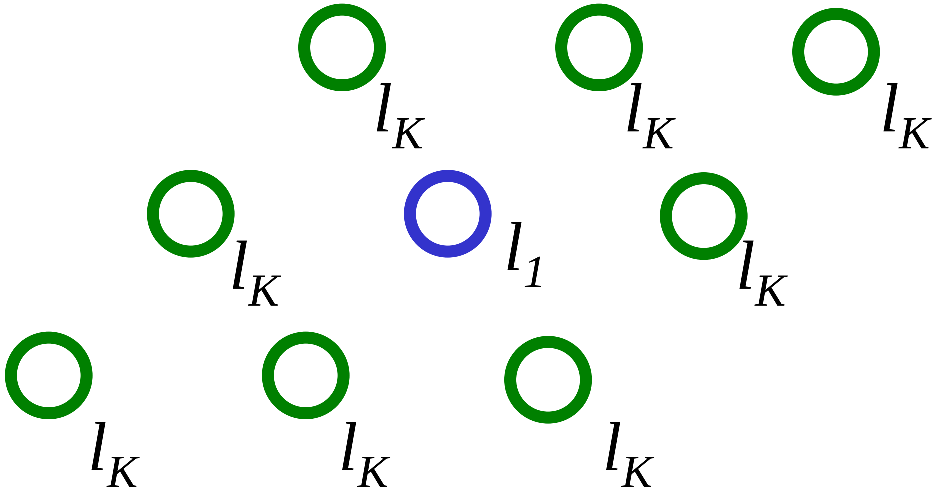
$$E(\mathbf{y}) = \sum_{i=1}^{N} E_i(y_i)$$

Model introduction

- Local classifiers

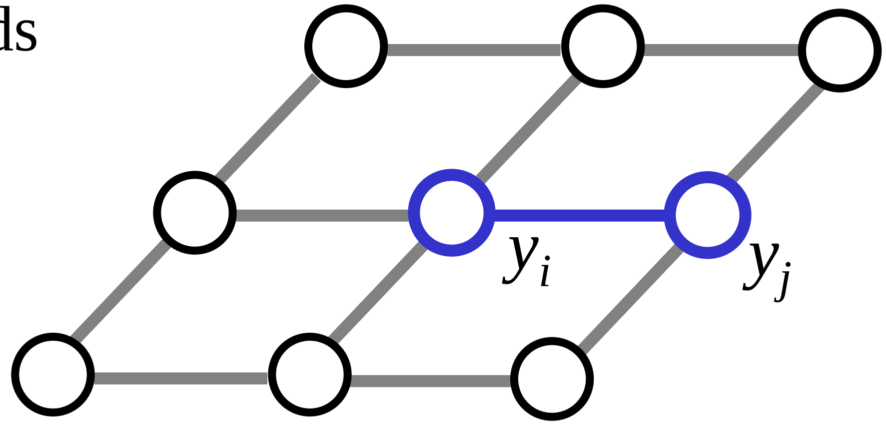
$$\mathbf{Y} = \{Y_1, \ldots, Y_N\}$$

$$Y_i \in \{\ell_1, \ldots, \ell_K\}$$



Model introduction

- Markov Random Fields



$$\mathbf{Y} = \{Y_1, \ldots, Y_N\}$$

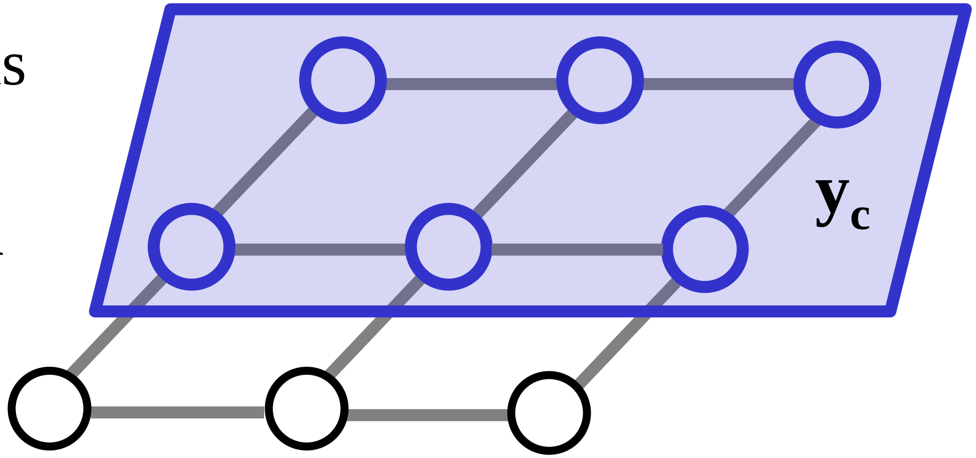
$$Y_i \in \{\ell_1, \ldots, \ell_K\}$$

$$E(\mathbf{y}) = \sum_{i=1}^{N} E_i(y_i) + \sum_{(ij)\in E} E_{ij}(y_i, y_j)$$

- Key concepts (see paper for details)
  - Each $E_c(\ )$ dependent on **features x** and **label**-specific **weights w**
  - **Classification**: optimal* labeling **y** can be found efficiently
    - ✓ [boykov-pami-01]
  - **Learning**: finding **w** is a convex optimization problem
    - ✓ [taskar-nips-03, ratliff-aistats-07]

Learning high-order interactions

- High-order interactions
  - [kohli-cvpr-07]
  - **Params not learned**



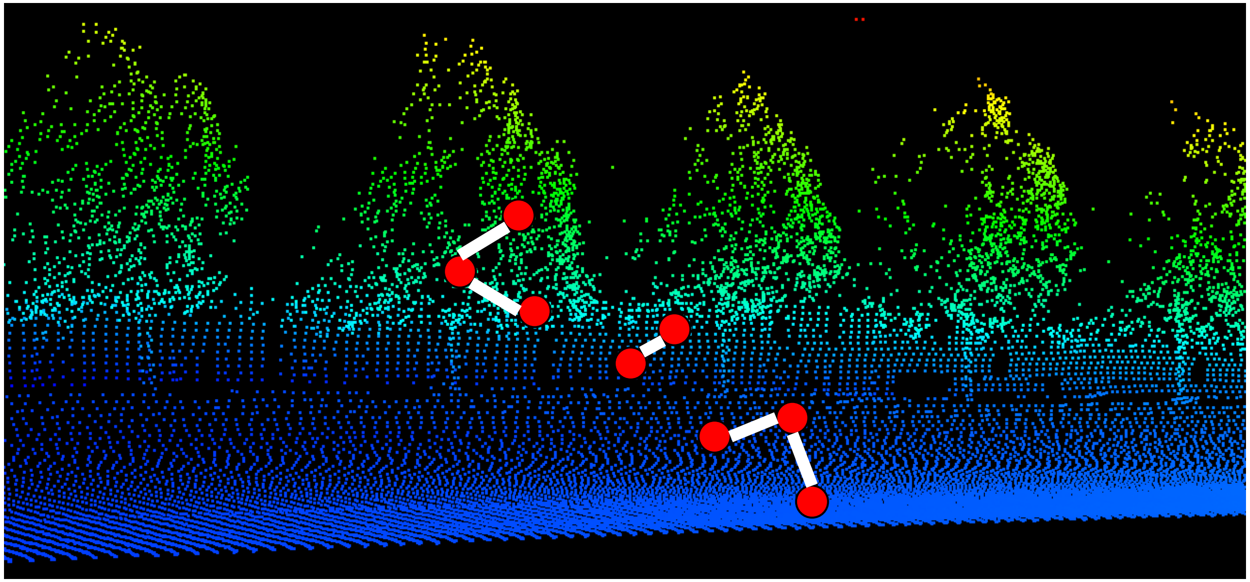
$$E(\mathbf{y}) = \sum_{i=1}^{N} E_i(y_i) + \sum_{(ij)\in E} E_{ij}(y_i, y_j) + \sum_{c\in S} E_c(\mathbf{y_c})$$

- **This work**: cast E_c under the same learning framework

$$E_c(\mathbf{y_c}) = \begin{cases} \mathbf{w_c^k} \cdot \mathbf{x_c} & \text{if } \forall i \in c, y_i = l_k \\ 0 & \text{otherwise,} \end{cases}$$

Context approximation

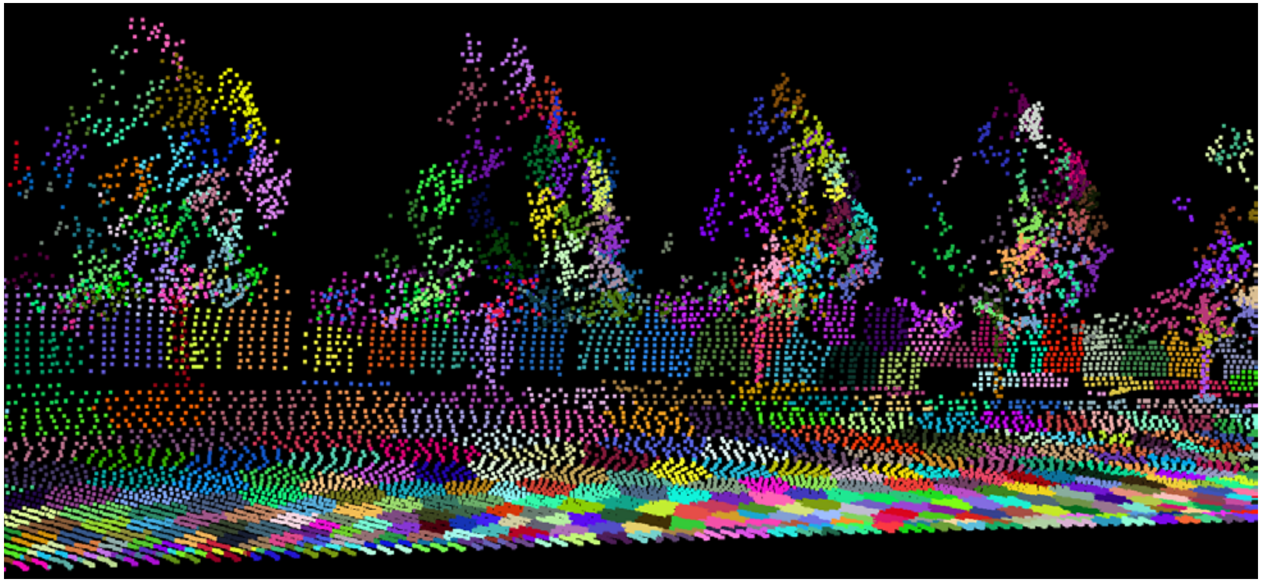
- Are pairwise interactions necessary?



(Edge construction = k-NN)

Context approximation

- Are pairwise interactions necessary?



(High-order clique construction = k-means clustering)

$$E_c(\mathbf{y_c}) = \begin{cases} \mathbf{w_c^k} \cdot \mathbf{x_c} & \text{if } \forall i \in c, y_i = l_k \\ 0 & \text{otherwise,} \end{cases}$$

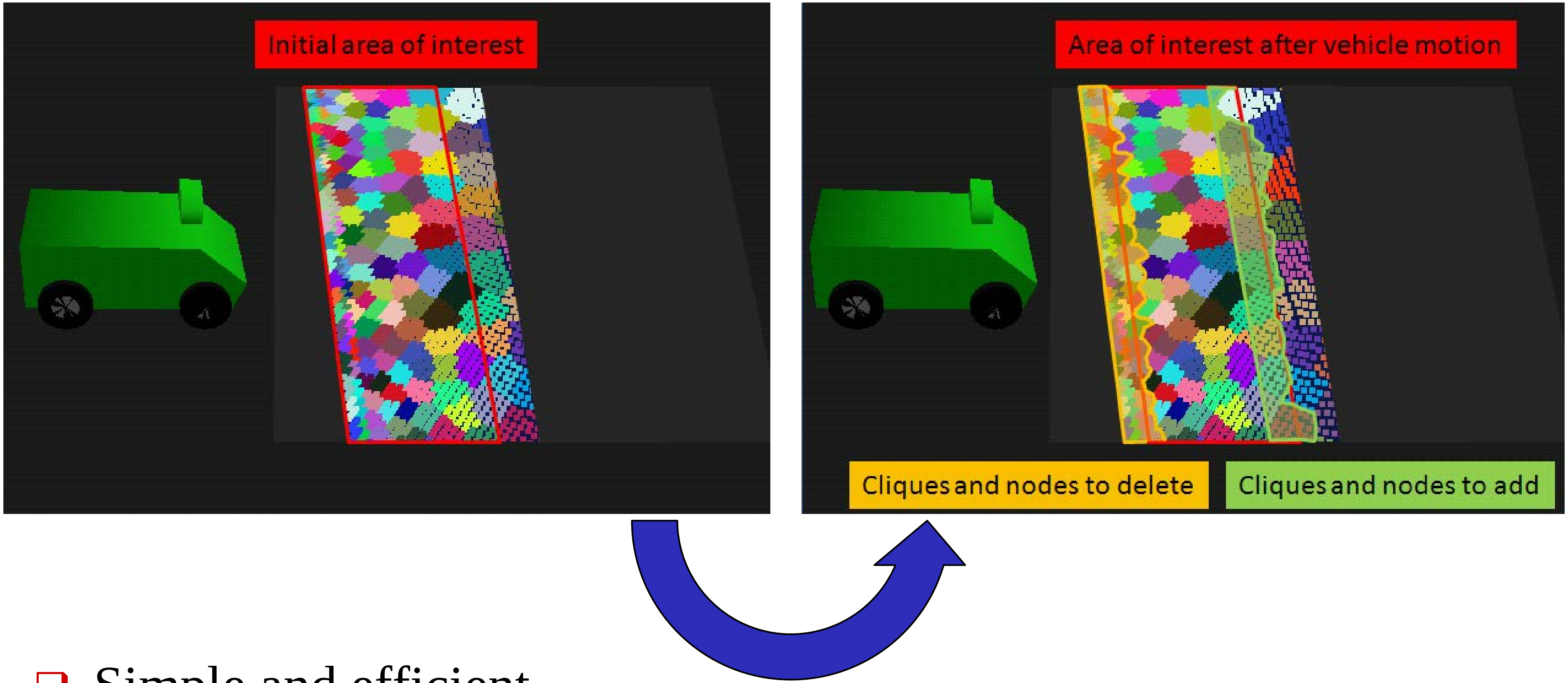
Counter-intuitive:
High-order inference is fast

- Classification comparison **vs** k-NN pairwise model
  - 1.2 M ground truth points

vs	Accuracy rate	Computation speedup (off-board)
5-NN	Slightly worse (87% vs 89%)	10x faster
3-NN	Similar (87% vs 88%)	2x faster

Onboard Classification

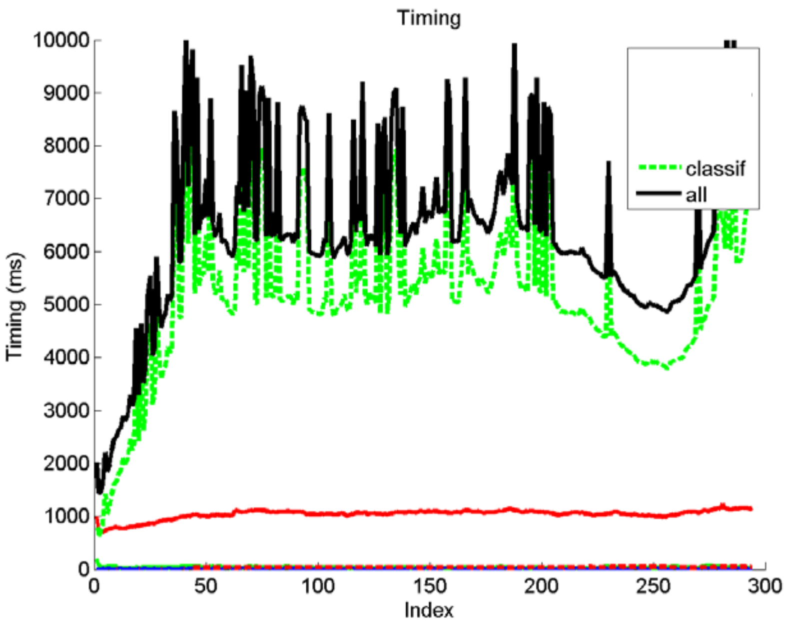
- Dynamic random field structure



- Simple and efficient

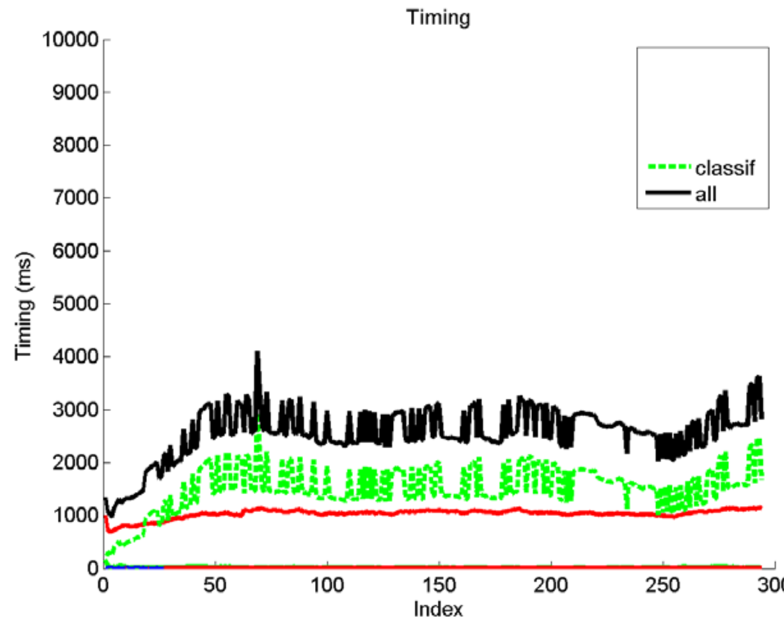
Onboard verification

- Comparison



Pairwise (3-NN)

Better

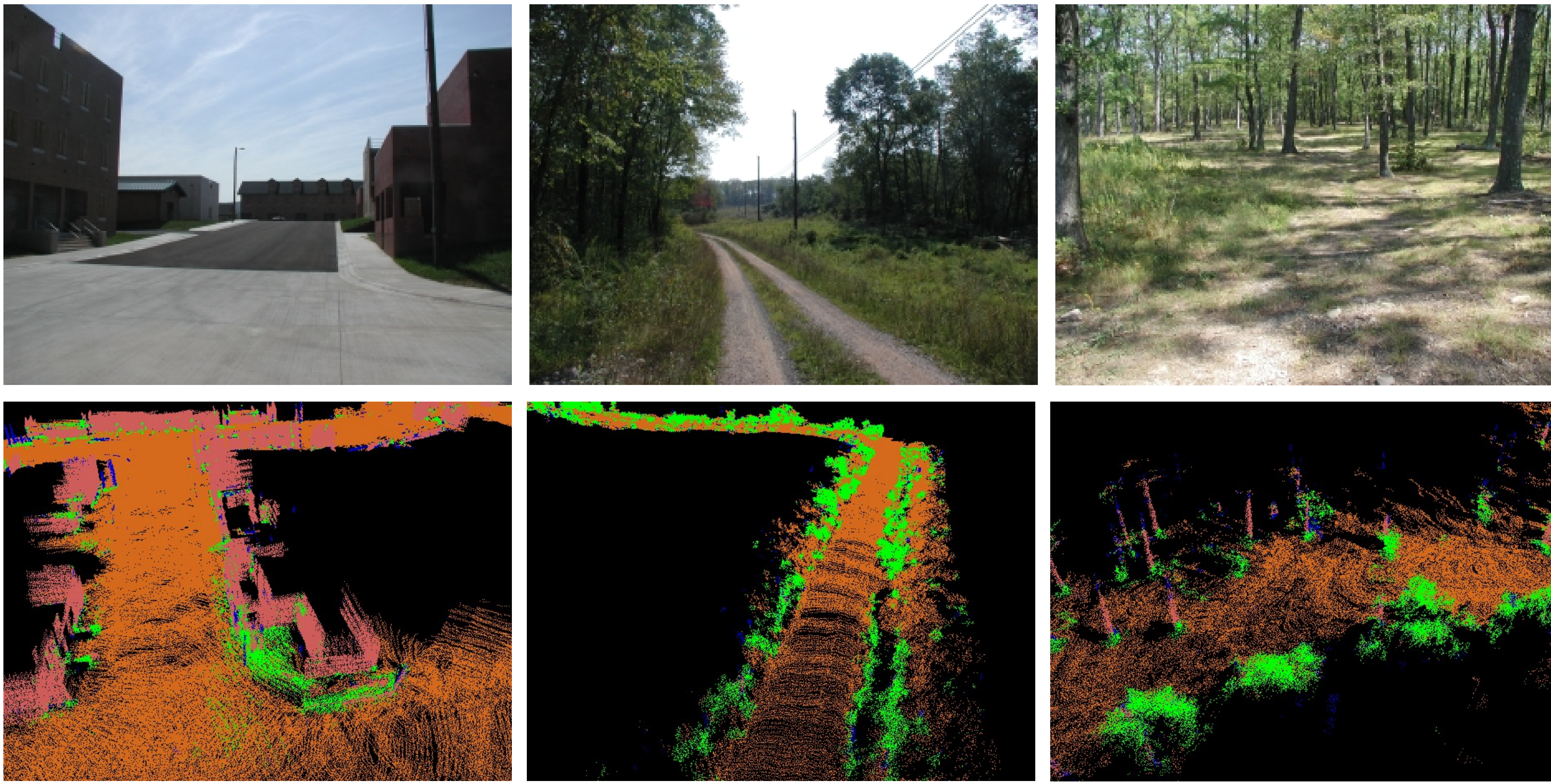


Proposed

  - **Green** = Classification
  - **Black** = Total processing time (green + updating graph structure)
- Onboard speedup: **3x**

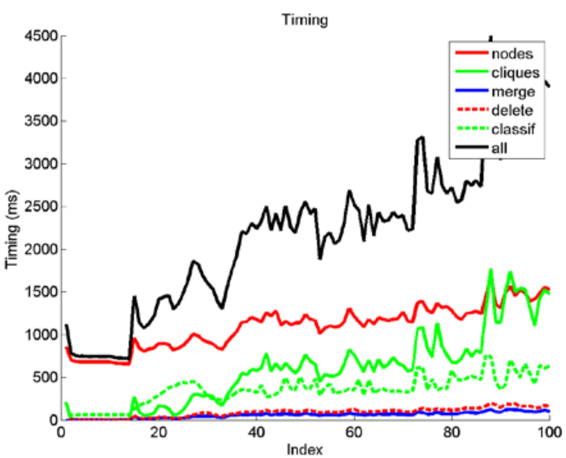
Field experimentation

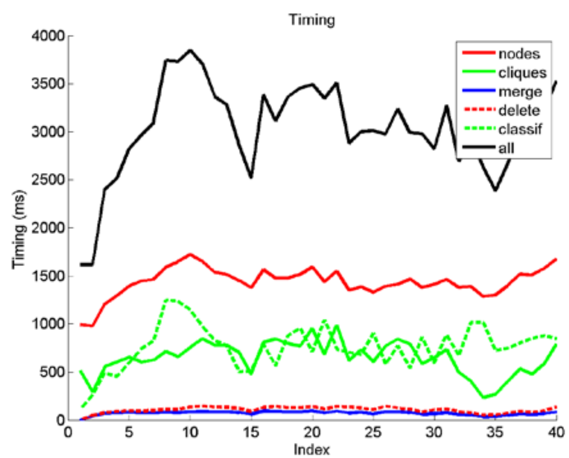
- Tested over 20 km of terrain, 25 x 50 m map
- Urban (MOUT), trail and forest environment

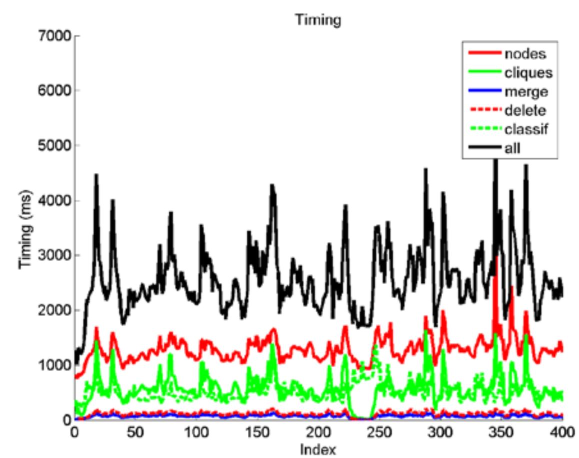


- Efficient onboard feature computation [lalonde-ijrr-07]

Field experimentation

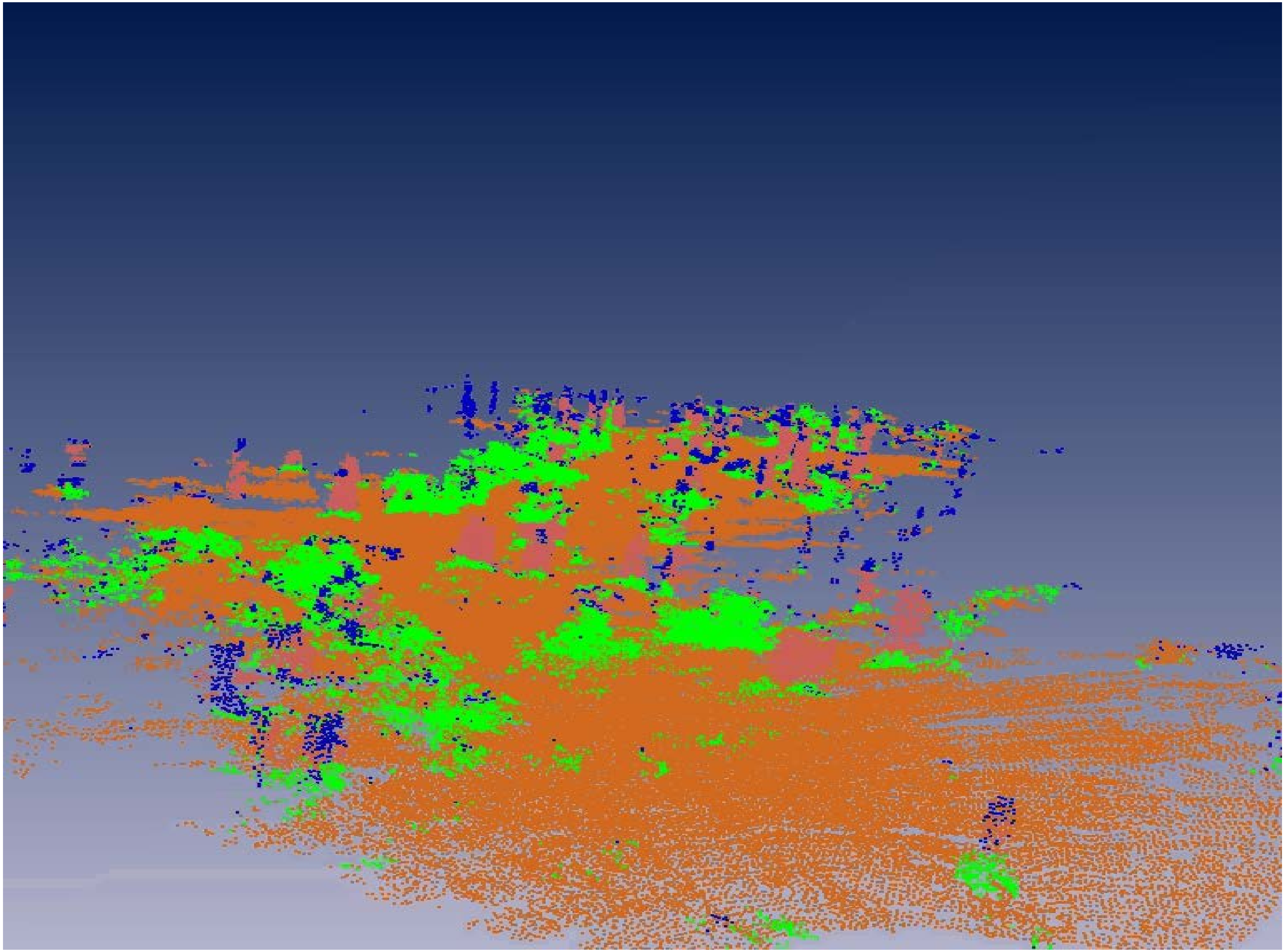






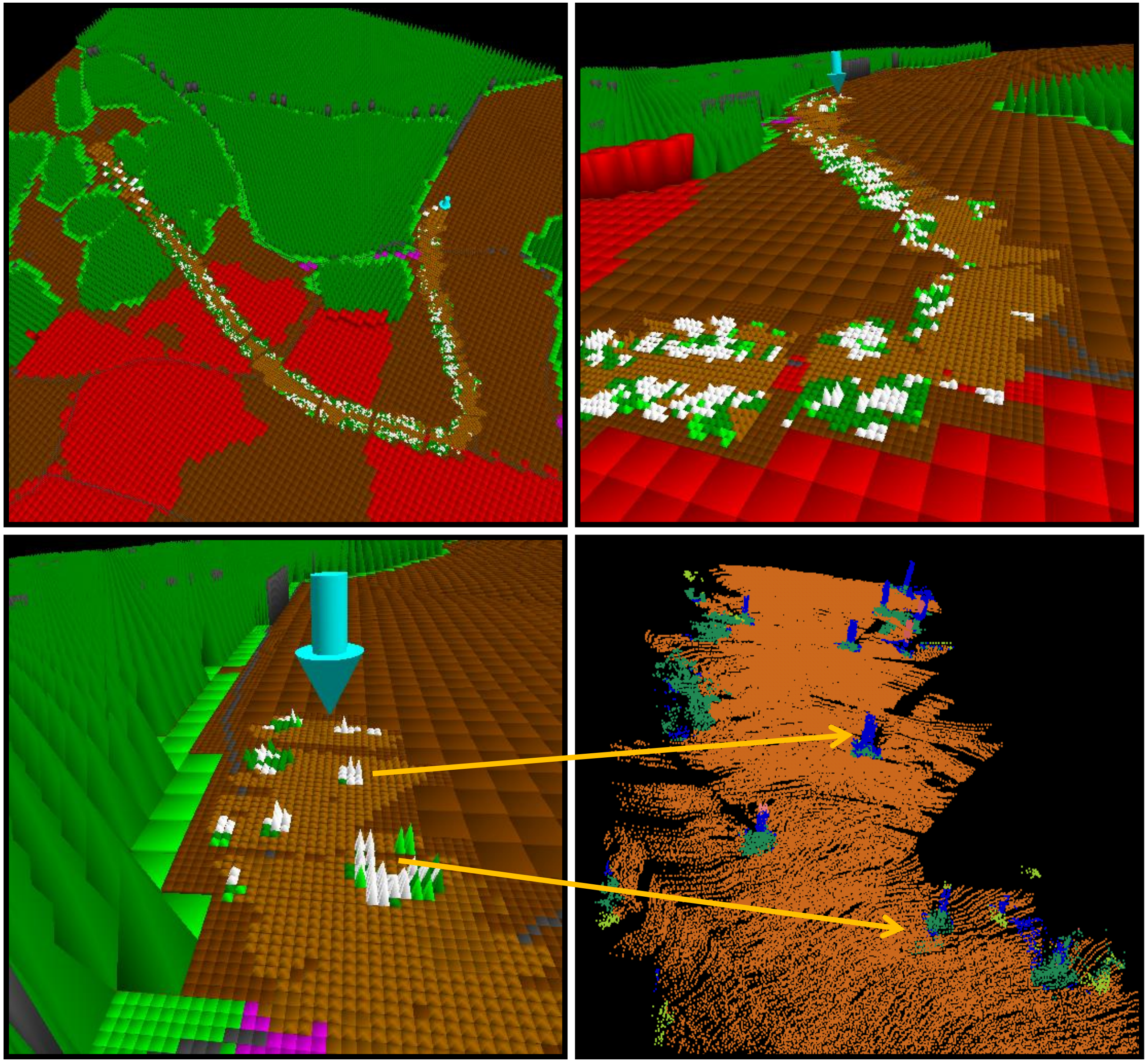
- Average speed: ~2 m/s

Forest Environment



Example of integration

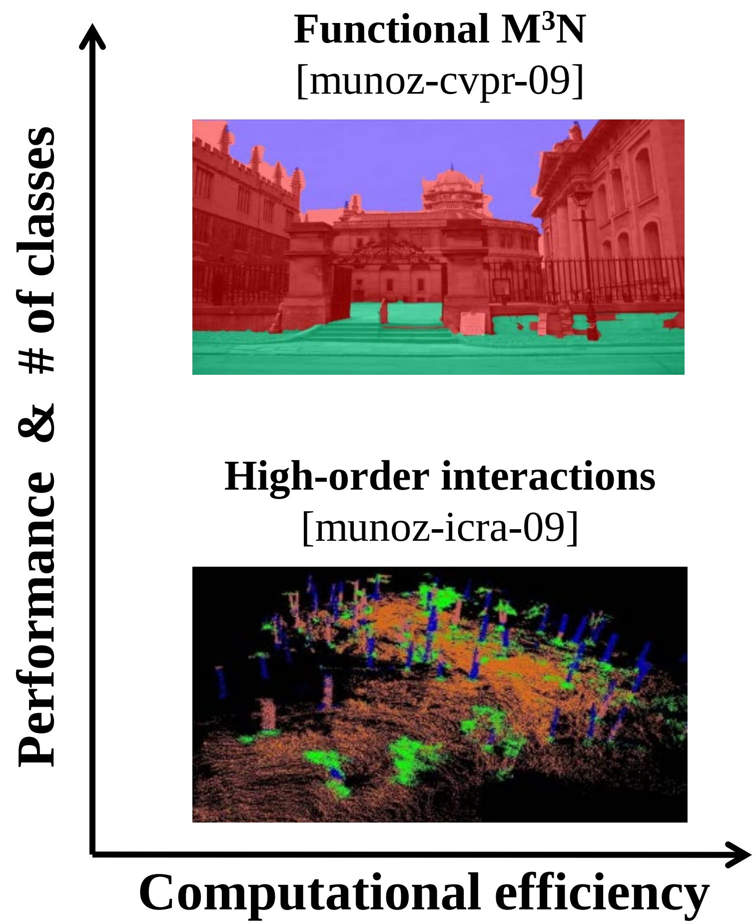
- Updating prior map for long range planning



Conclusion

- **Contributions**
  - Efficiently learn high-order interactions
  - Context approximation for onboard processing
    - ✓ Fast
    - ✓ Works well in practice

- **Limitations**
  - Computation time
  - Clique interactions
  - Optimization



Thank you

- Acknowledgements
  - U.S. Army Research Laboratory
  - General Dynamics Robotic Systems