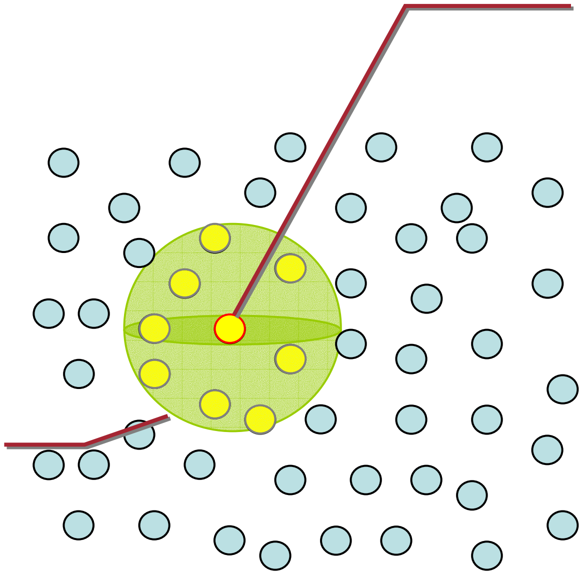
Improving 3-D processing: an efficient data structure and scale selection

Jean-François Lalonde
Vision and Mobile Robotics Laboratory

Goal

- Improve 3-D signal processing techniques
  - Speed of execution
  - Accuracy
- Rely on local computations



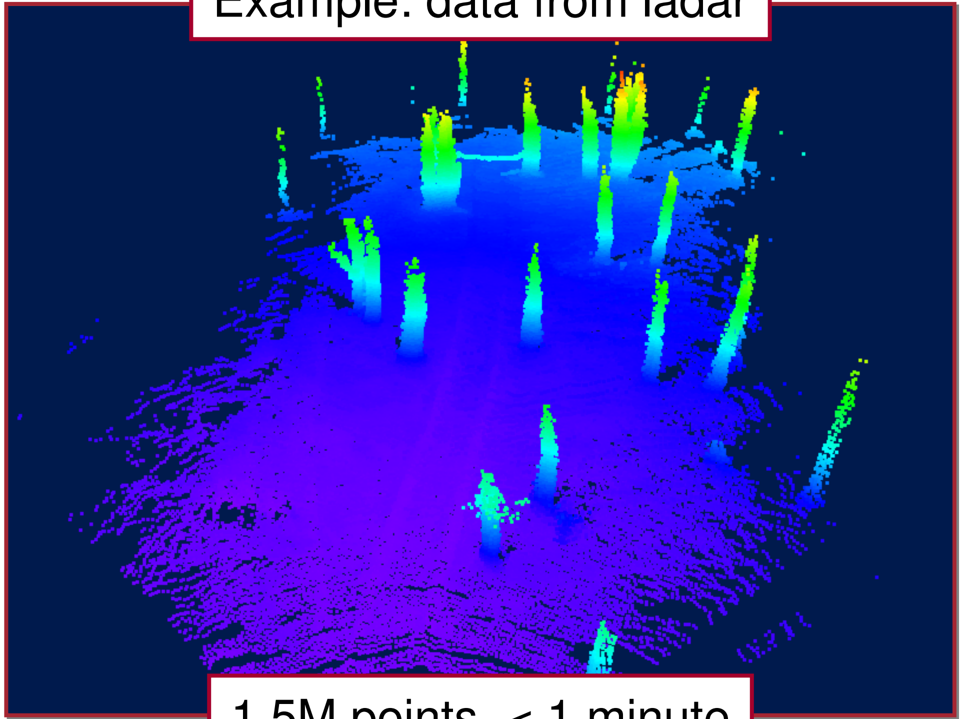
Point of interest
Scan through every point in the dataset

Support region
Local computations on highlighted points

3-D signal processing challenges

- Very large amount of data
  - > 1,000,000 points
- Dynamic
  - Data arrives sequentially
    - No bounds *a priori*
  - Very high data rate (~100,000 points/sec)
- Varying density & geometry
  - Empty space
  - Holes, discontinuities, junctions

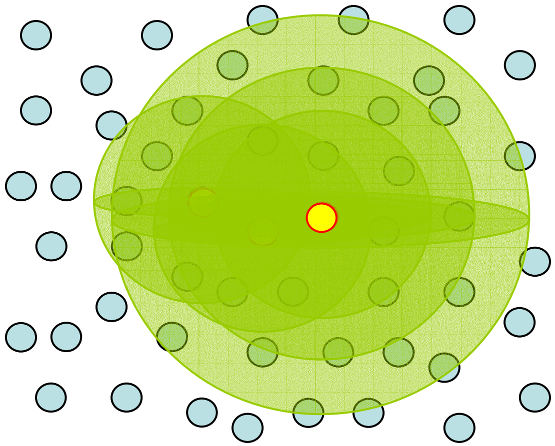
Example: data from ladar



1,5M points, < 1 minute

Challenges & proposed solutions

- Local computations
  - Very tedious → need to be fast
  - How to define the size?



- Proposed solutions
  - Improve the speed → efficient data structure
  - Define the size → explore scale selection in 3-D

Plan

- Example application
  - Ground robot mobility

- Efficient data structure
  - Approach
  - Experimental results

- Scale selection problem
  - Overview
  - Experimental results

Plan

- **Example application***
  - **Ground robot mobility**

- Efficient data structure
  - Approach
  - Experimental results

- Scale selection problem
  - Overview
  - Experimental results

* Originally introduced in CTA project [Vandapel04, Hebert03]

Example: perception for robot mobility

- Developed a framework
  - Enables navigation in variety of complex environments

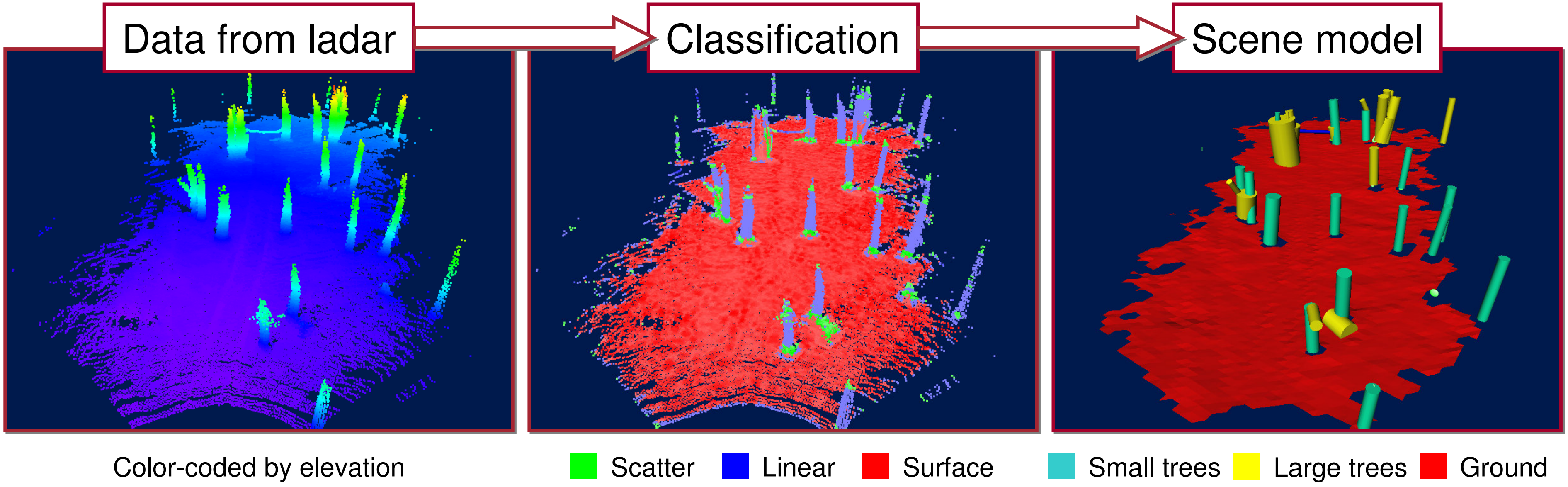
  - 3-D representation is necessary
    - Previous approaches (2-D) insufficient
- Based on
  - Local feature extraction
  - Classification

Perception for robot mobility (contd.)

- GDRS eXperimental Unmanned Vehicle
- GDRS Mobility LADAR
  - Time-of-flight
- Mounted on turret
- 100,000 3-D points per second

Perception for robot mobility (contd.)

- Voxelize data
  - Store sufficient statistics
- Compute local PCA features
  - Eigenvalues of local covariance matrix
- Perform on-line classification
  - Mixture of gaussians to model feature distributions
- Grouping & modeling



Plan

- Example application
  - Ground robot mobility

- **Efficient data structure***
  - **Approach**
  - **Experimental results**

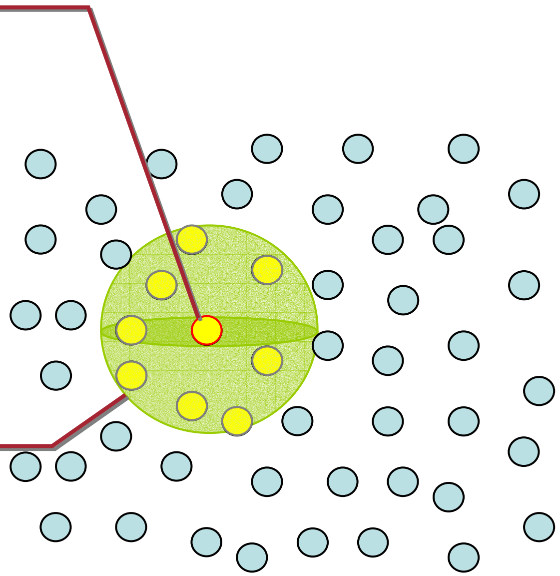
- Scale selection problem
  - Overview
  - Experimental results

* Jointly with N. Vandapel and M. Hebert

Local computation on 3-D point sets

Point of interest
Scan through all points in the dataset

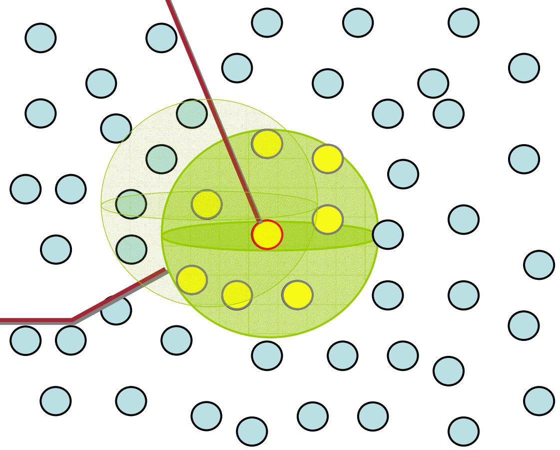
Support region
Local computation on highlighted points



Local computation on 3-D point sets

Point of interest
Scan through all points in the dataset

Support region
Local computation on highlighted points

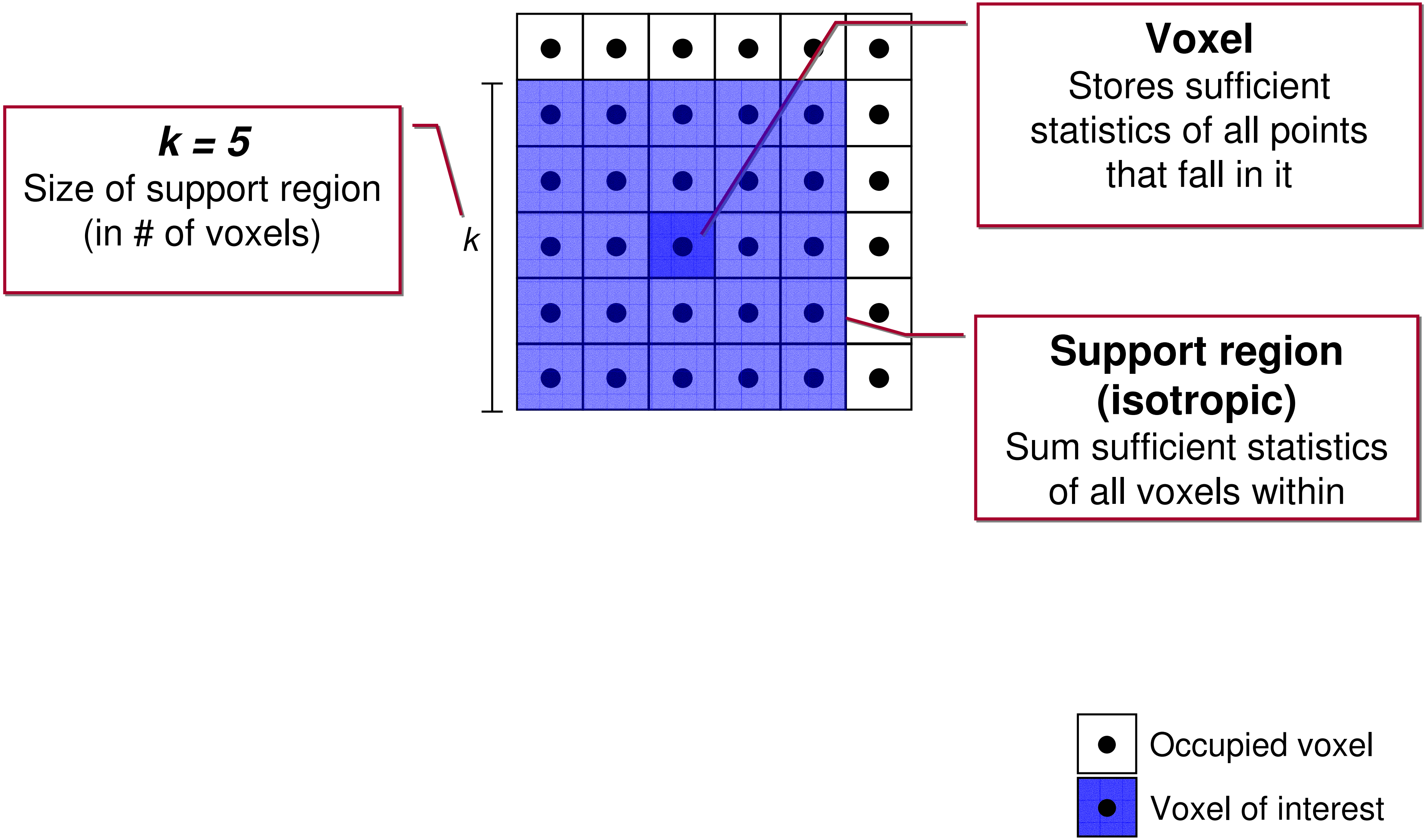


Very expensive, but can reuse data from overlapping regions

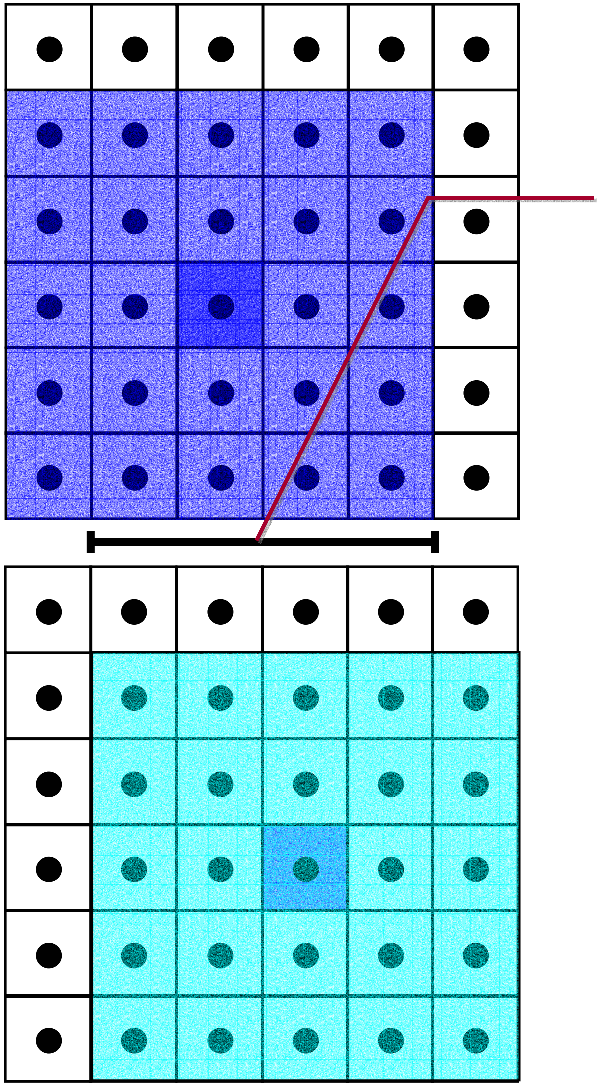
Challenges

- Nature of data
  - Ladar on a moving platform [Lacaze02]
    - Dynamic (accumulation)
  - Need to process data continuously
- Efficient operations
  - Insertion and access
  - Range search
    - Local computations
- Traditional techniques do not apply
  - Tree-based data structures [Samet81, Liu04, Gray04]
    - Suitable for static and high-dimensional data

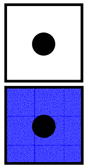
Concept – 2-D example



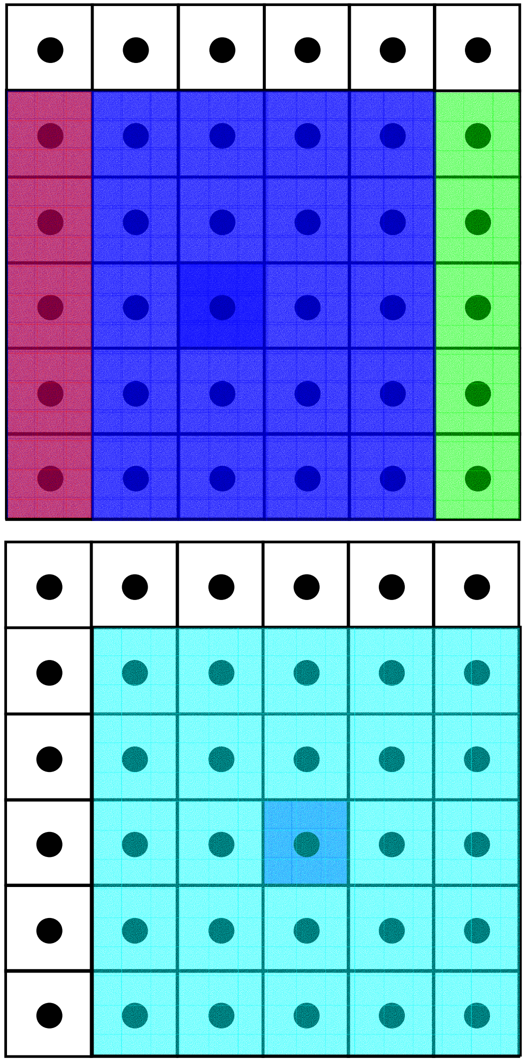
Concept – 2-D example



Overlap
How can we reuse pre-computed data?

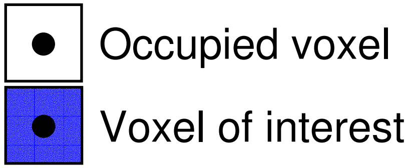


Concept – 2-D example

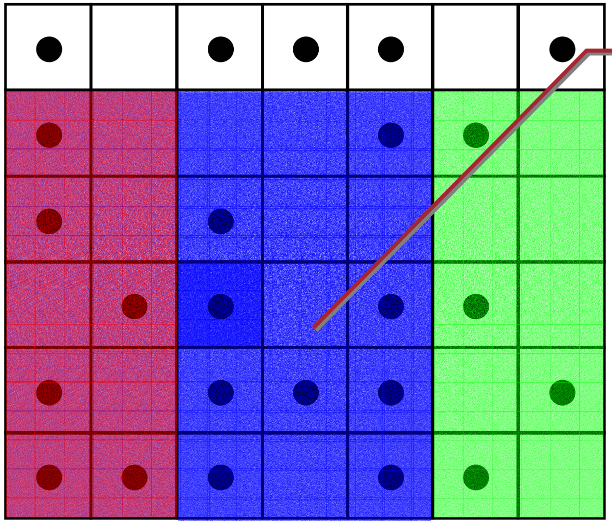


1. Start with the blue region
2. Add the green column
3. Subtract the red column

- Proven to be efficient in image processing [Faugeras93]
- Challenge in 3-D: data is sparse

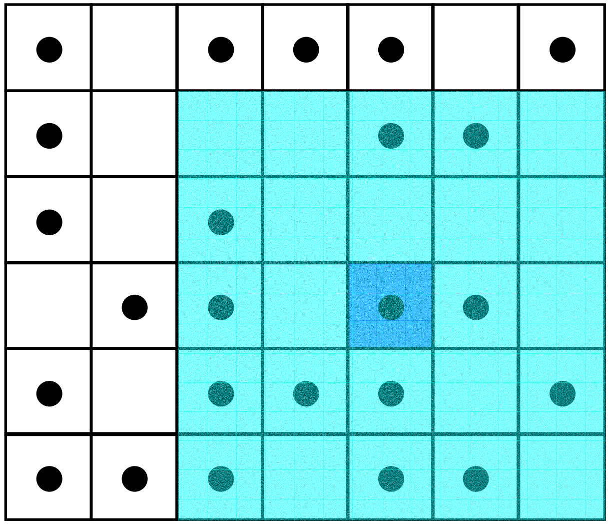


2-D example, sparse data

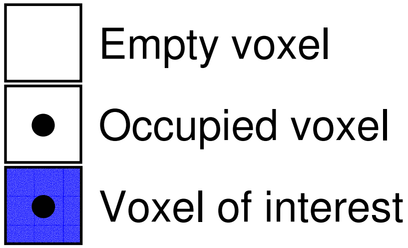


Sparse data
Some voxels are empty

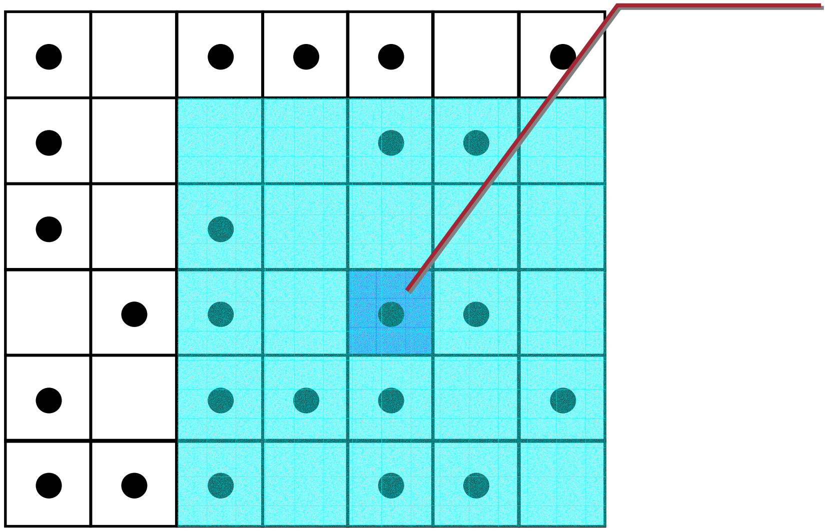
1. Start with the blue region
2. Add the green columns
3. Subtract the red columns



May not always be useful to reuse data



2-D example, sparse data

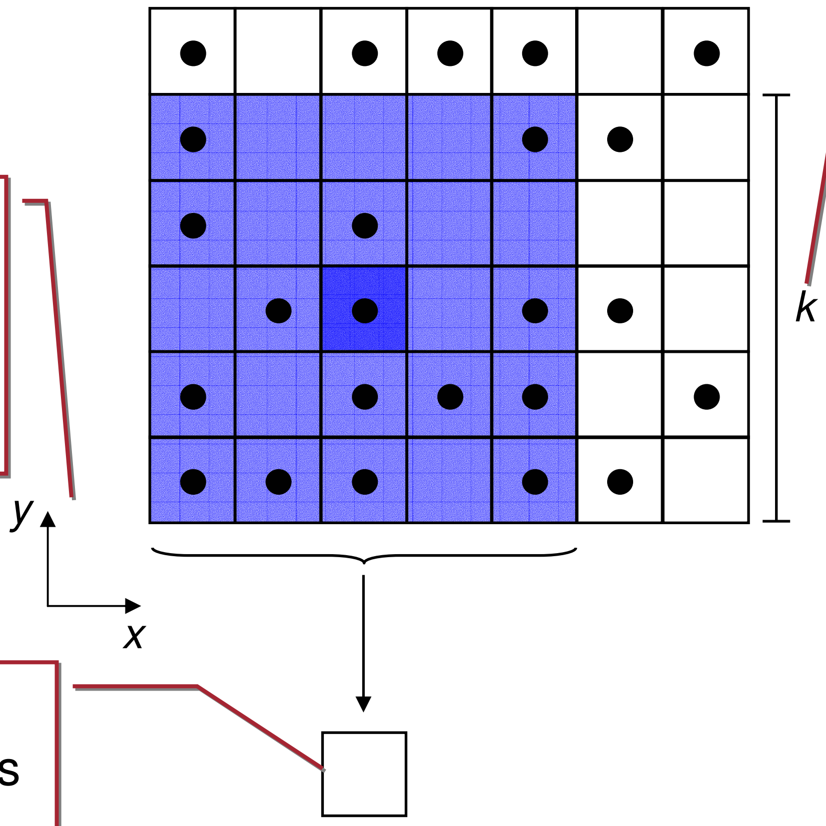


Where is the previous result?

- 2 approaches:
  - Default scan
  - Optimized scan

Approach 1: default scan

1. Scanning direction
Example: *x* first
Arbitrary

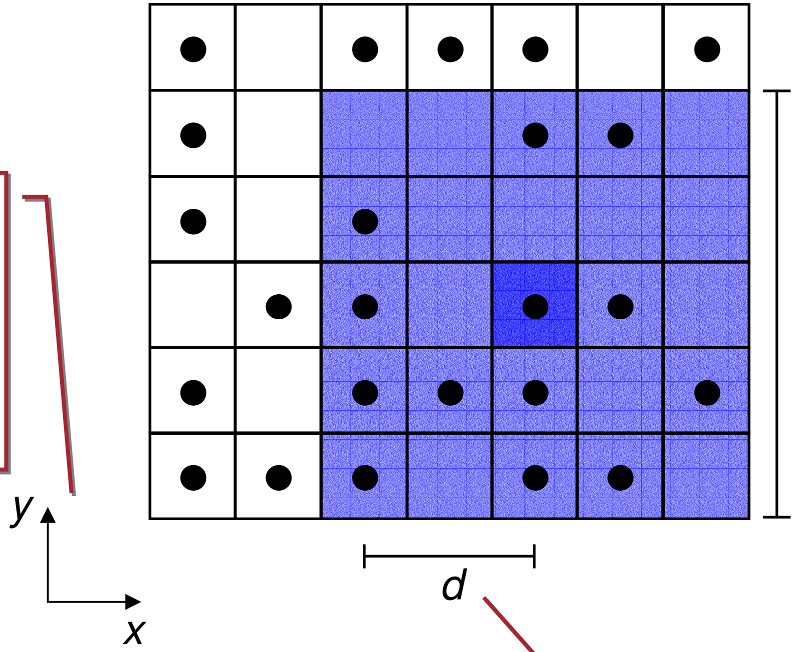


k = 5
Size of support region
(in # of voxels)

2. Memory
Compute partial sums
and store result &
location in memory

Empty voxel
Occupied voxel
Voxel of interest

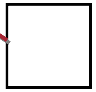
Approach 1: default scan



1. Scanning direction
Example: *x* first
Arbitrary

***k* = 5**
Size of support region
(in # of voxels)

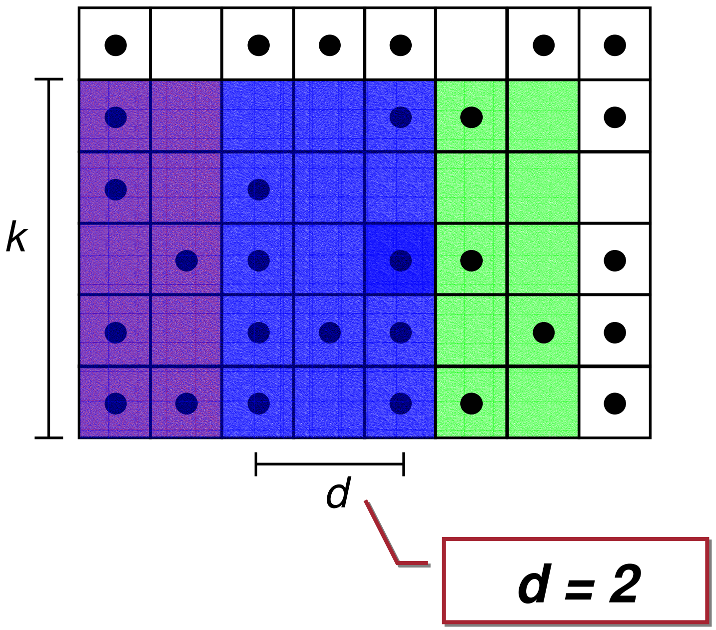
2. Memory
Compute partial sums
and store result &
location in memory



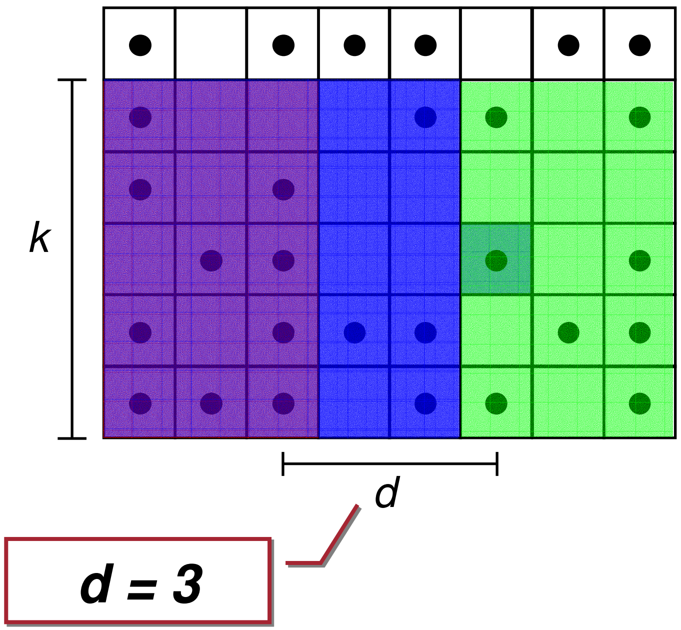
***d* = 2**
Distance between interest
voxel and previous result
(in # of voxels)

2 cases

$$d < \frac{k}{2}$$



$$d > \frac{k}{2}$$

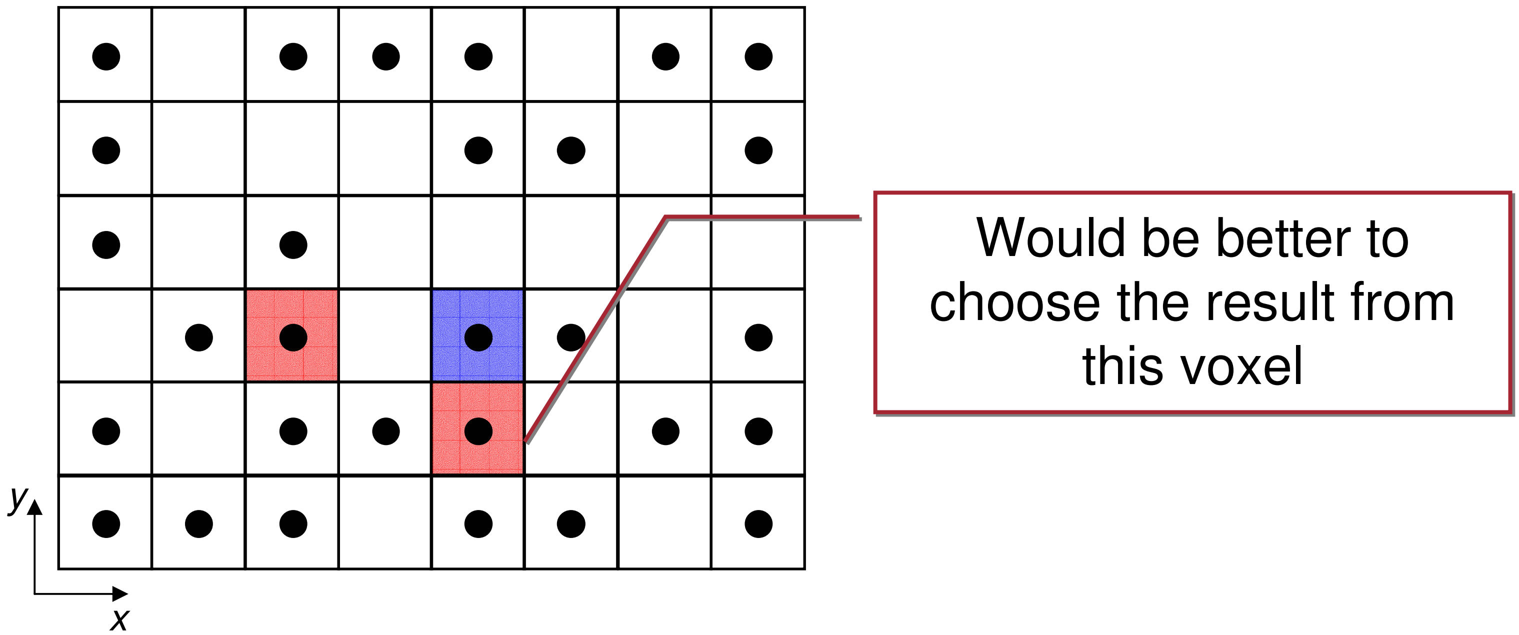


Reuse previous results

Do not reuse, recompute

Approach 2: optimized scan

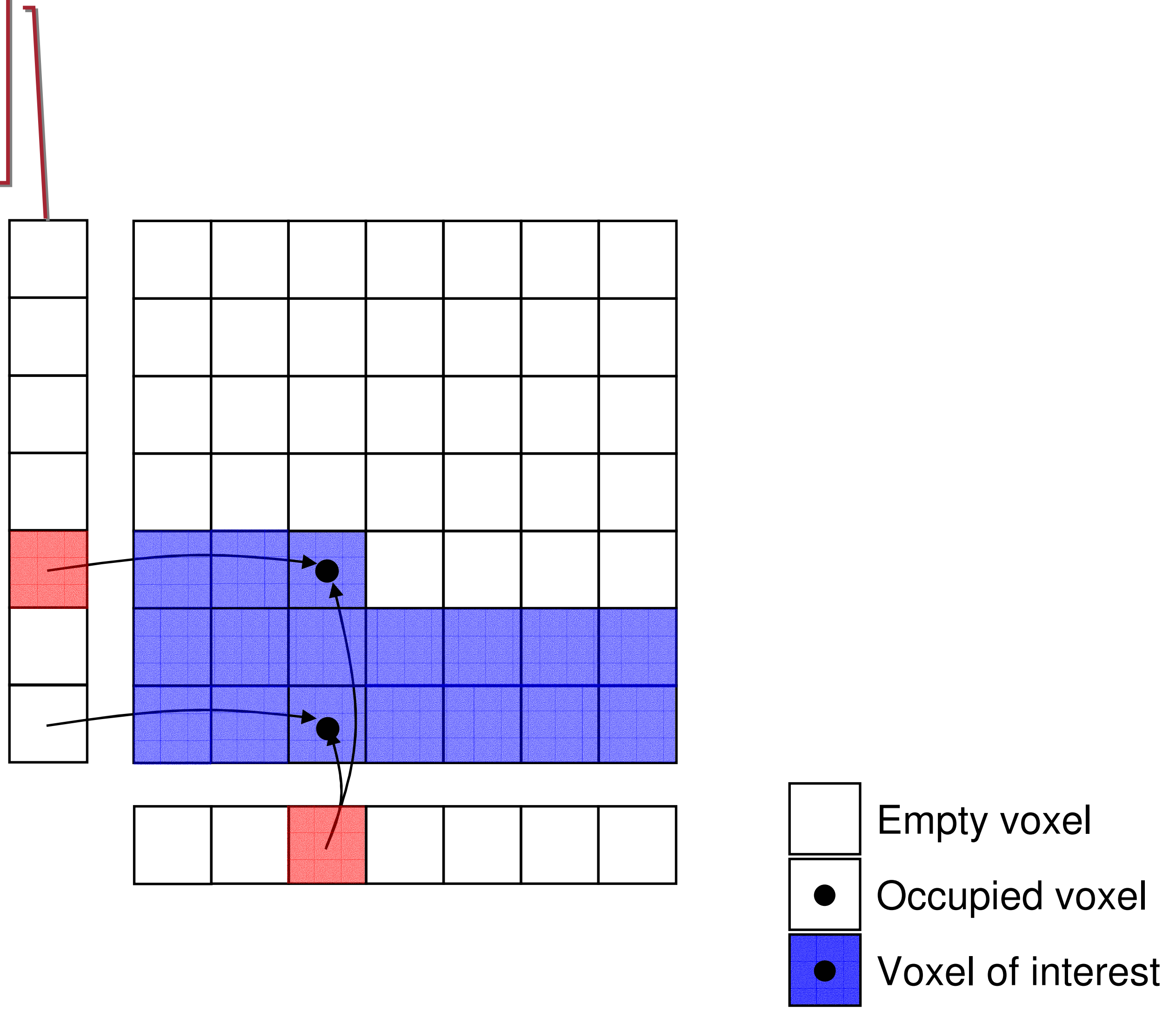
- Can we do better?



- Choose closest (along *x*, *y* or *z*)

Approach 2: optimized scan

Additional arrays
Store all previous results & locations

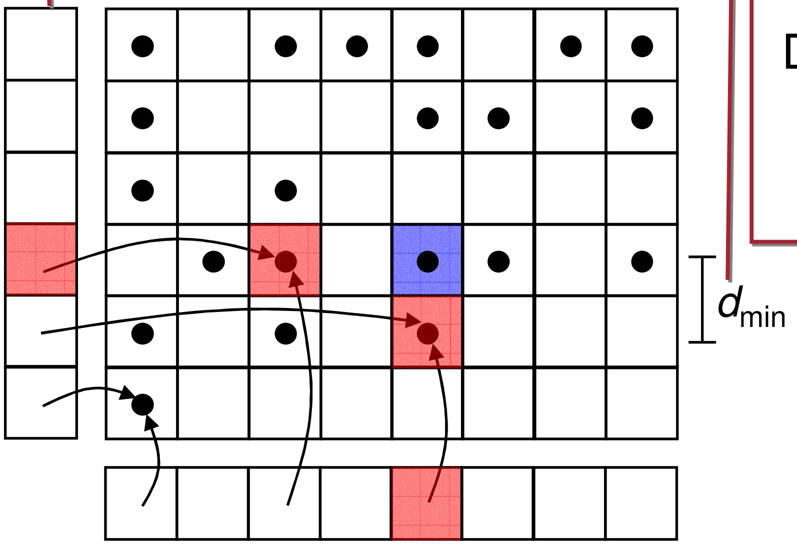


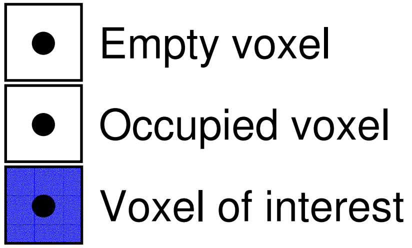
Approach 2: optimized scan

Additional arrays
Store all previous results & locations

d

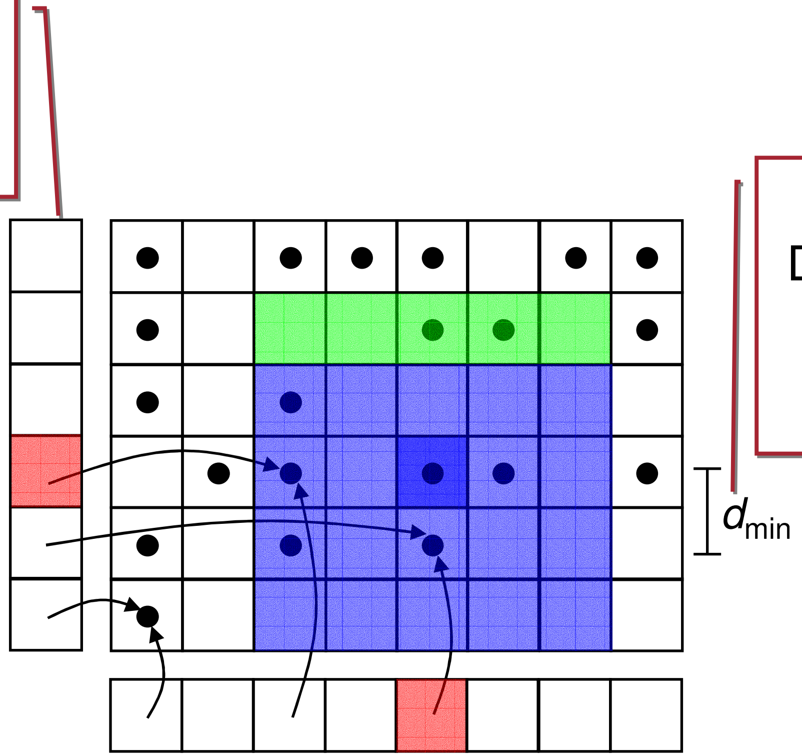
d_{min}
Distance between voxel of interest and closest previous result





Approach 2: optimized scan

Additional arrays
Store all previous results & locations



***d*$_{min}$**
Distance between voxel of interest and closest previous result

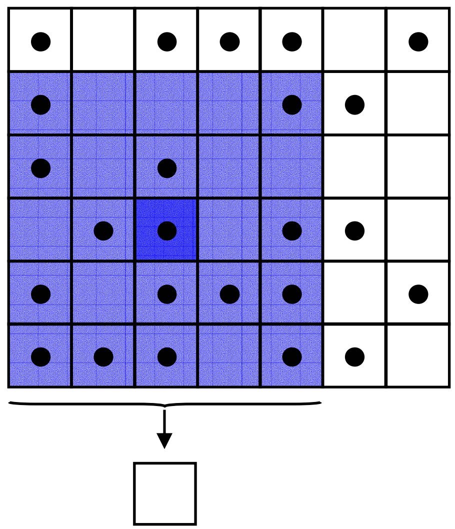
Reuse data if condition is met

$$d_{\min} < \frac{k}{2}$$

- Empty voxel
- Occupied voxel
- Voxel of interest

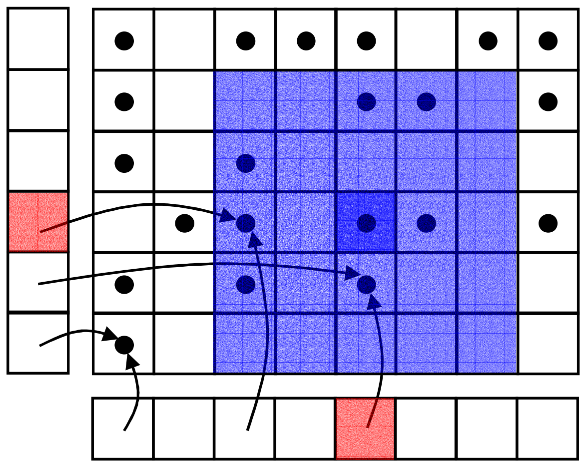
Comparison

Default scan



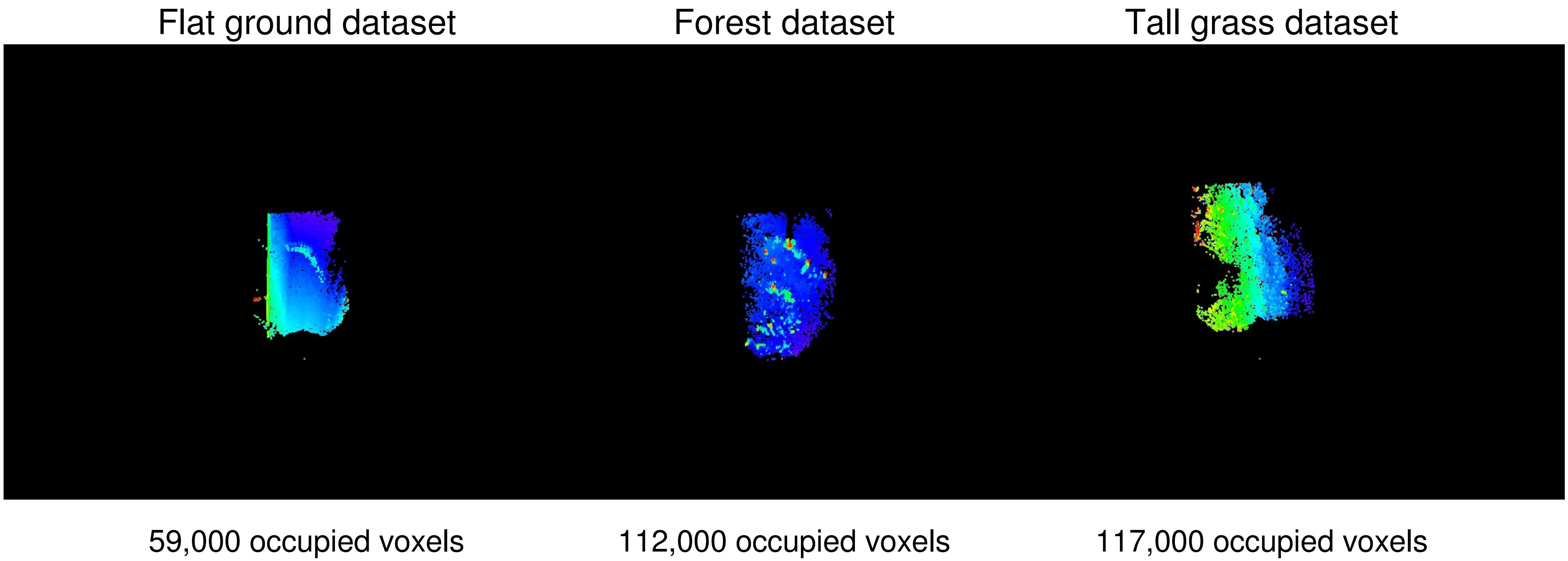
+ Very easy to implement
+ Minimal overhead
    one memory location
    one distance computation
- Dependent on scanning direction (user input)

Optimized scan



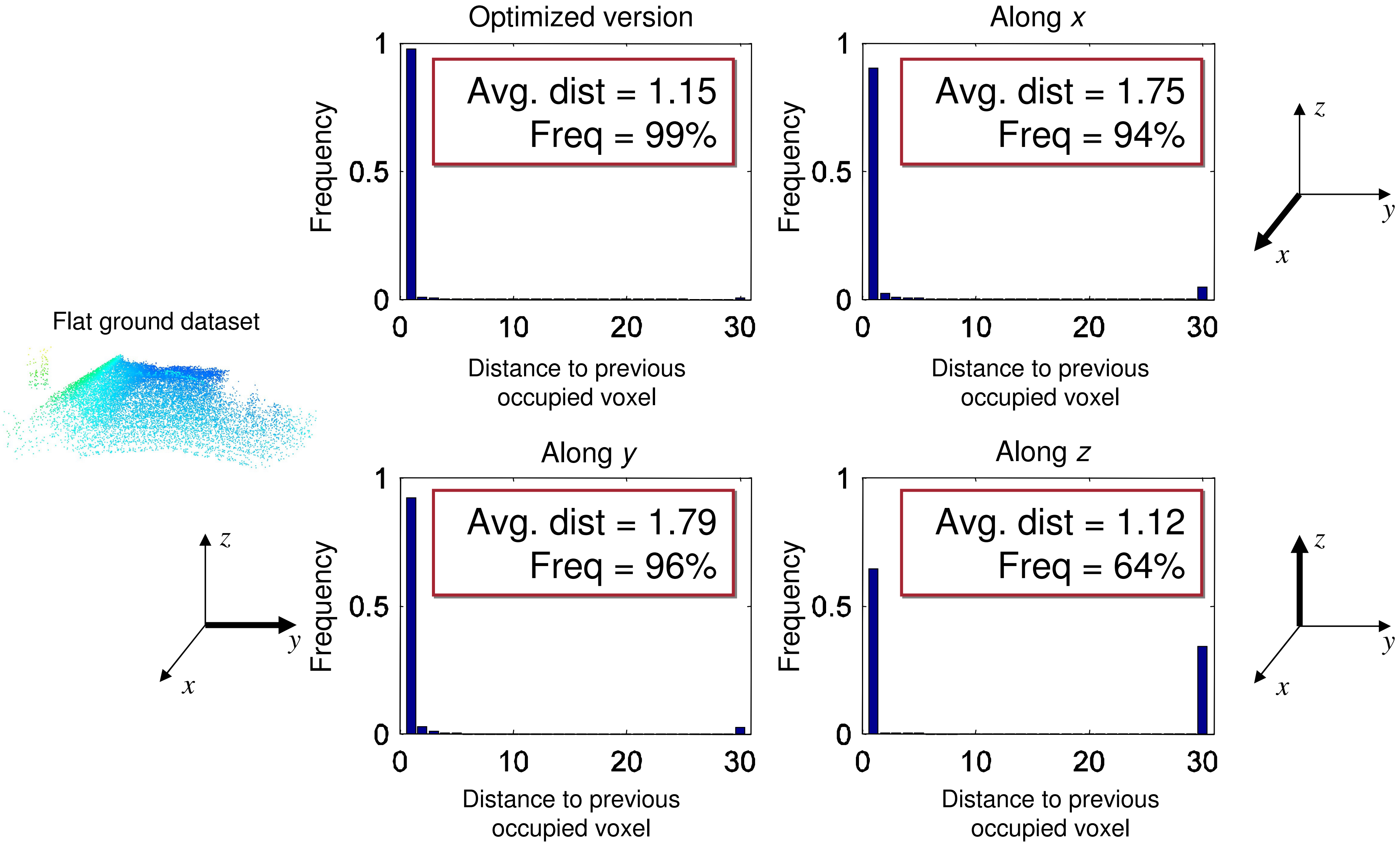
+ Independent on scanning direction
+ Provide highest speedup
- Harder to implement
    direction determined dynamically
- Additional overhead
    memory usage
    3 distance computations

Experiments - overview

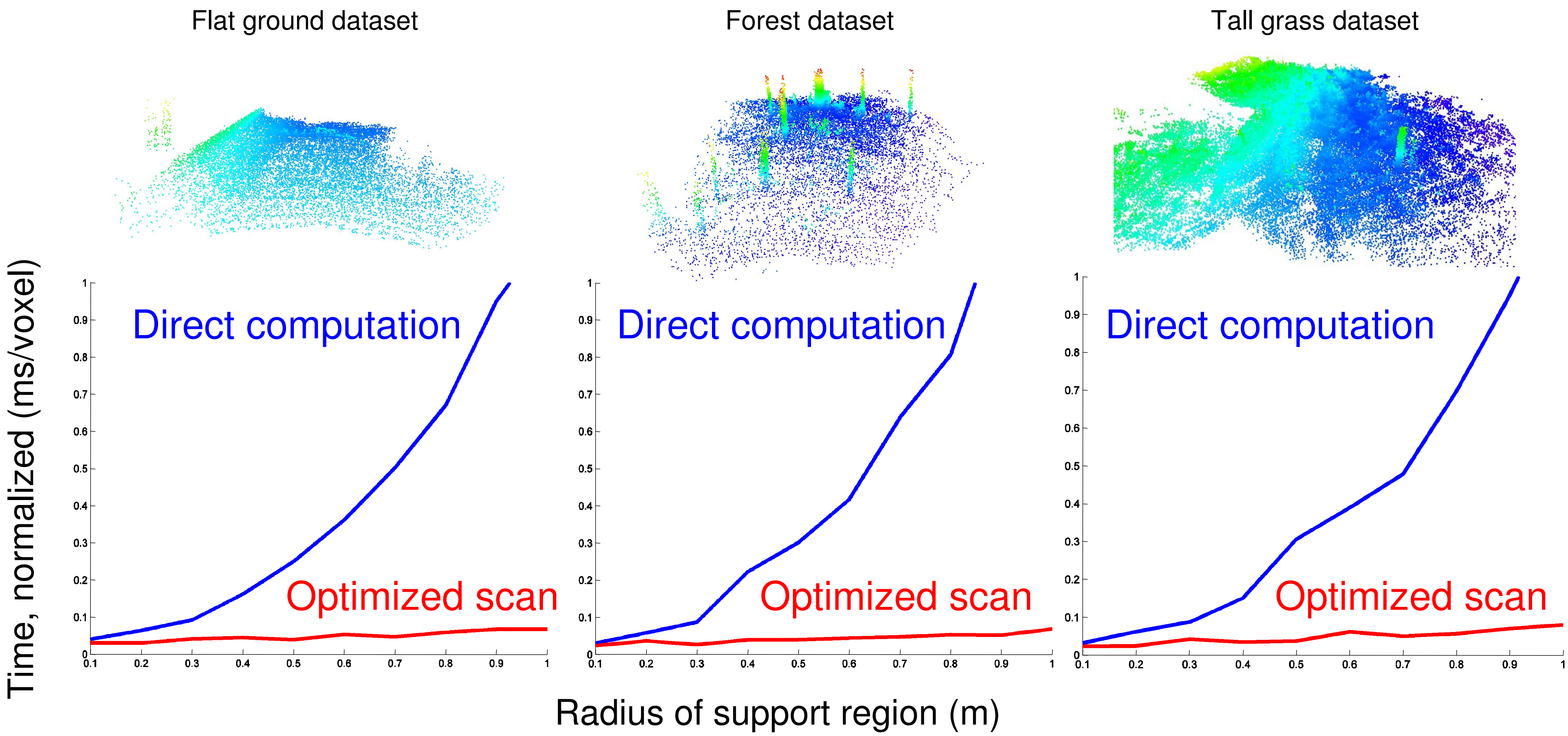


- Voxel size of 0.1m
- Experiments:
  - Influence of scanning direction
  - Speedup on different scenes
- Data collected by the robot
- Both batch & live playback data processing

Experiments – scanning direction



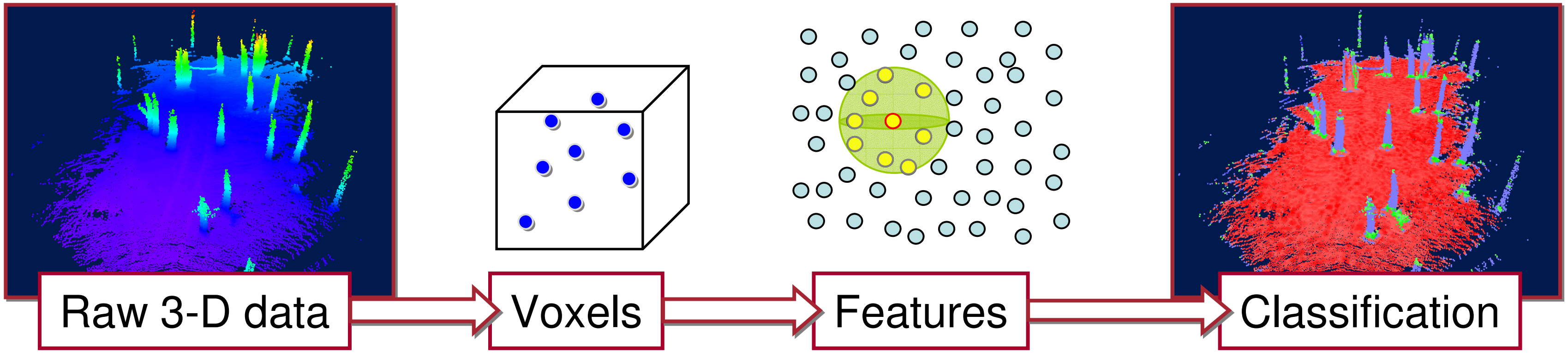
Experiments - speedup



- Speedup of 4.5x at radius of 0.4m (k = 9)

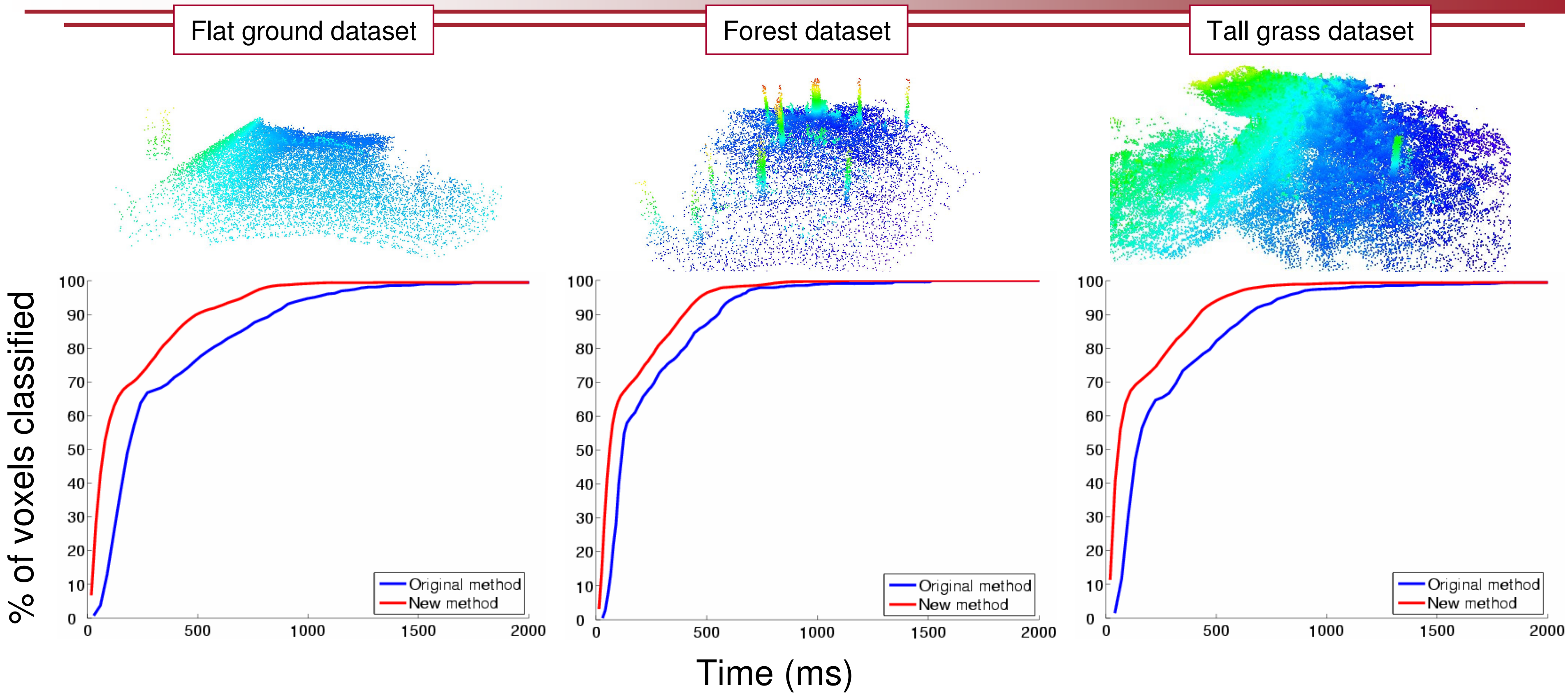
Experiments – dynamic data

- Batch timing definition not suitable
  - Closely related to application
- New definition
  - Tied to obstacle detection
  - Time between voxel creation and classification

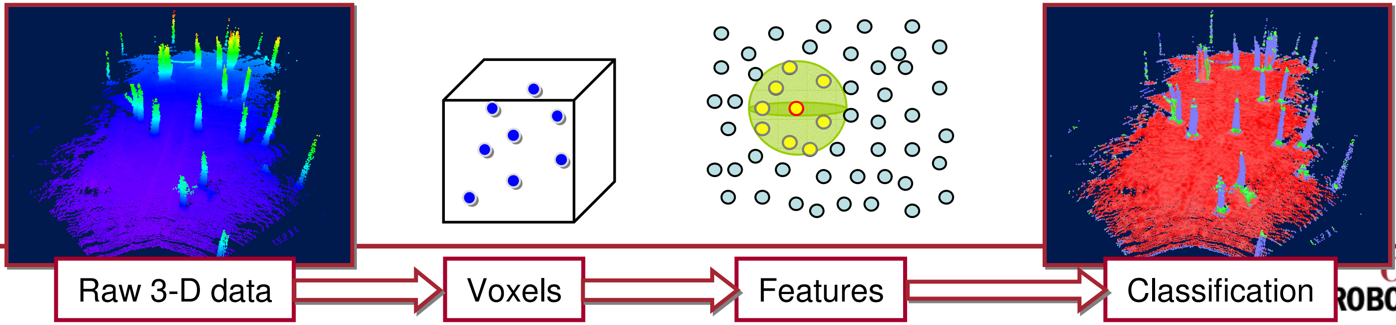


  - Cumulative histograms
- Playback results

Experiments – dynamic data (contd.)



- Time to classify 90% of voxels: 40% improvement in speed



Data structure – summary

- Summary
  - Data structure with corresponding approach to speedup full 3-D data processing
  - Example in context of classification
  - 4.5x speedup for 3-D range search operation
  - Robot: ~100m @ 1.5m/s → ~8km @ 5m/s
- Limitations
  - Trade-off: hard to evaluate a priori
    - Gain of reusing data
    - Memory and processing overhead of more complex methods
- Future work
  - Other uses
    - Different steps in processing pipeline

Plan

- Example application
  - Ground robot mobility

- Efficient data structure
  - Approach
  - Experimental results

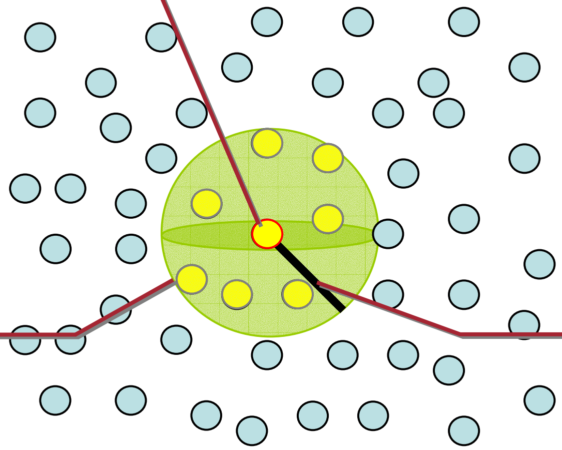
- **Scale selection problem***
  - **Overview**
  - **Experimental results**

* Jointly with R. Unnikrishnan, N. Vandapel and M. Hebert

Problem

- Find best estimate of the normal at a point

Point of interest

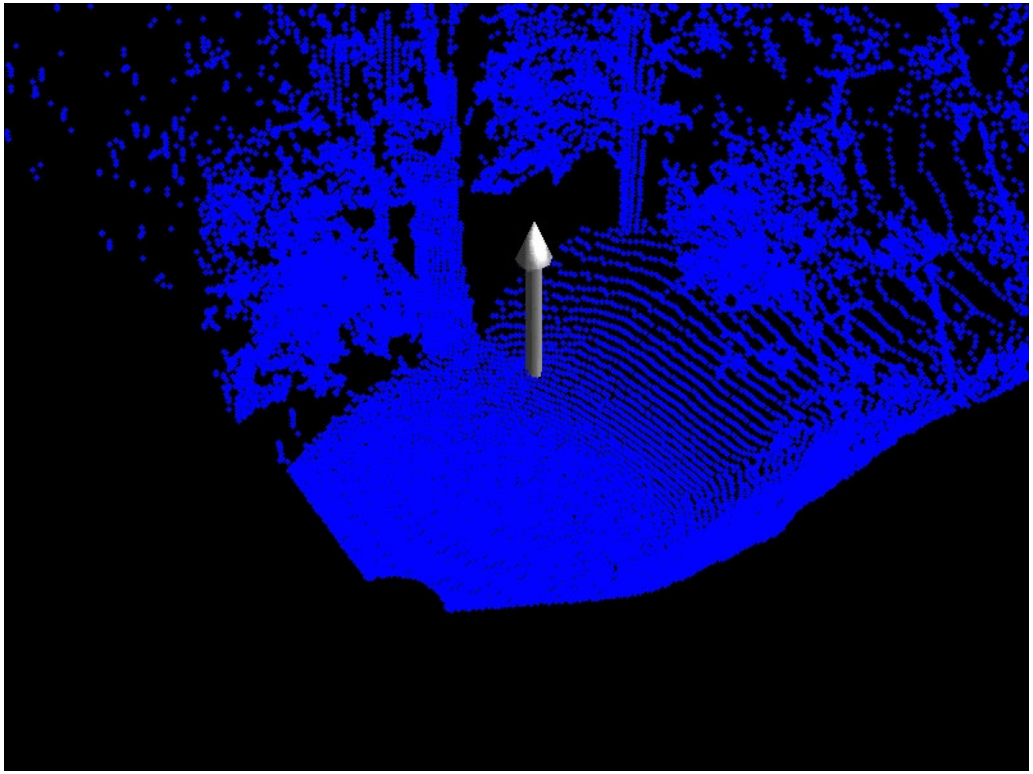


Support region
Fitting of some sort

Scale
Radius of support region

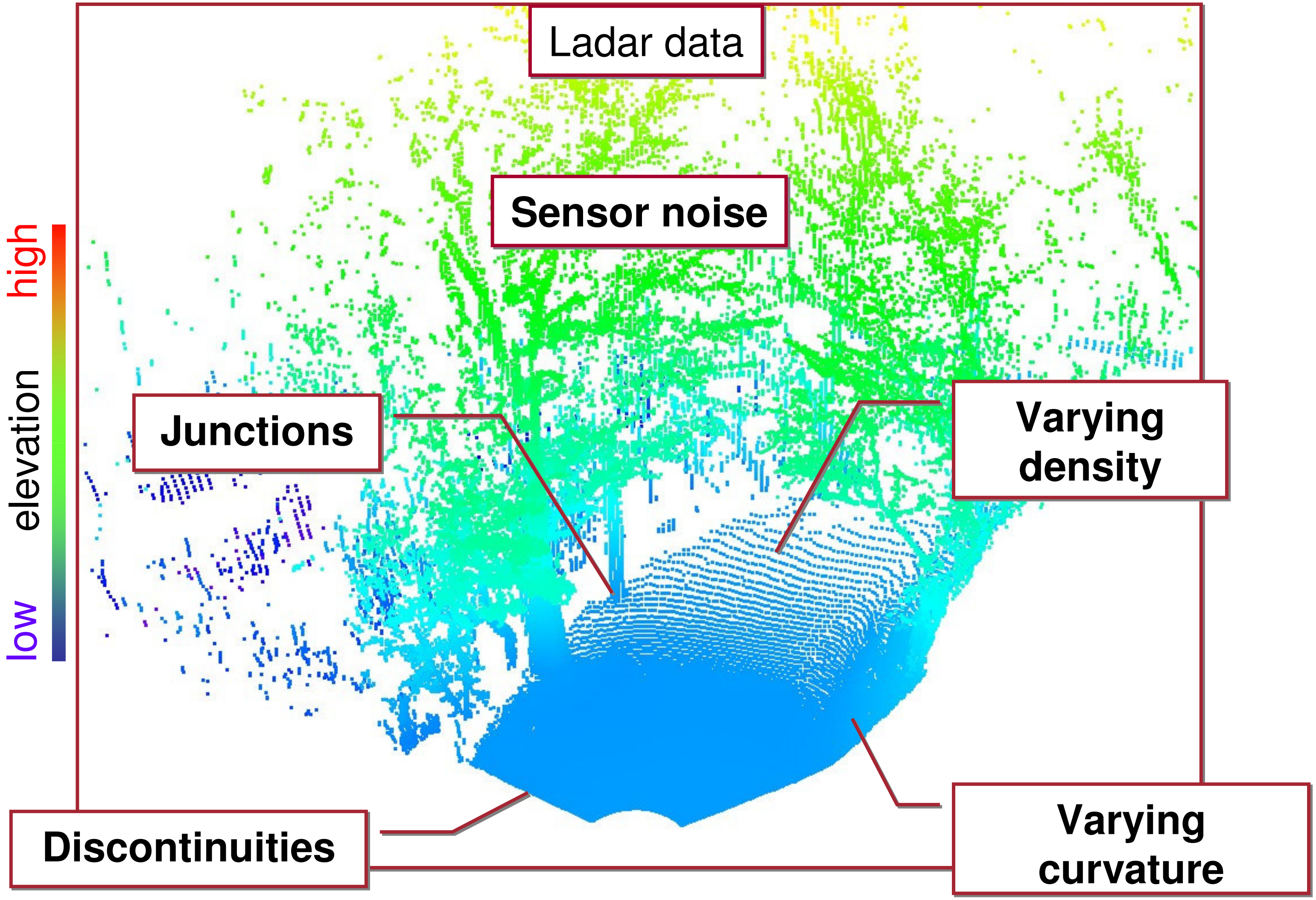
- **Best** normal → **Best** scale!

What is the best support region size?



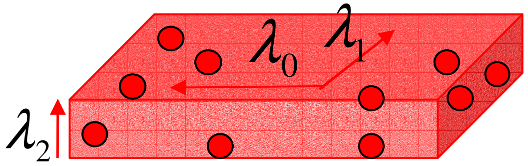
- Scale theory well-known in 2-D [Lindeberg90]
- No such theory in 3-D, ad-hoc methods
  - [Tang04a]: Tensor voting: no relation between region size and classification
  - [Pauly03]: Lines, no theoretical guarantees, no generalization for surfaces
  - [Tang04b]: Lines, fitting at increasing scales

Problem: challenges



Approach

- Focus analysis to surfaces
  - Larger source of errors

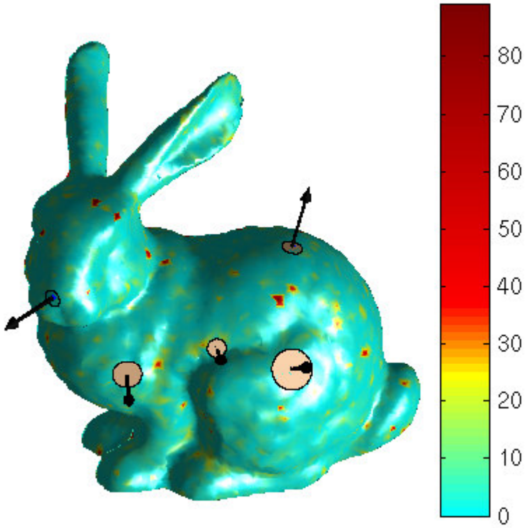


- Hypothesis

Optimal scale for **geometry**	→	Good feature for **classification**

Approach (contd.)

- Apply existing solution proposed to a different problem
  - Graphics community
  - [Mitra05]
    - Minimum spatial density (no holes)
    - No discontinuities
    - Small noise and curvature



dense, complete 3-D models

- Test our hypothesis

Optimal scale for **geometry** → Good feature for **classification**

- Present initial experimental results

[Mitra05] N. Mitra, A. Nguyen and L. Guibas, Estimating surface normals in noisy point cloud data. *Intl. Journal of Computational Geometry and Applications*, 2005.

Optimal scale selection for normal estimation [Mitra05]

- Analytic expression for optimal scale

$$r = \left(\frac{1}{\kappa} \left(\theta_3 \frac{\sigma_n}{\sqrt{\varepsilon\rho}} + \frac{\theta_4}{2} \sigma_n^2 \right) \right)^{\frac{1}{3}}$$

r
Estimated scale

κ
Estimated local curvature*

ρ
Estimated local density

σ_n
Sensor noise

known

unknown

* Curvature estimation from [Gumhold-01]

Algorithm

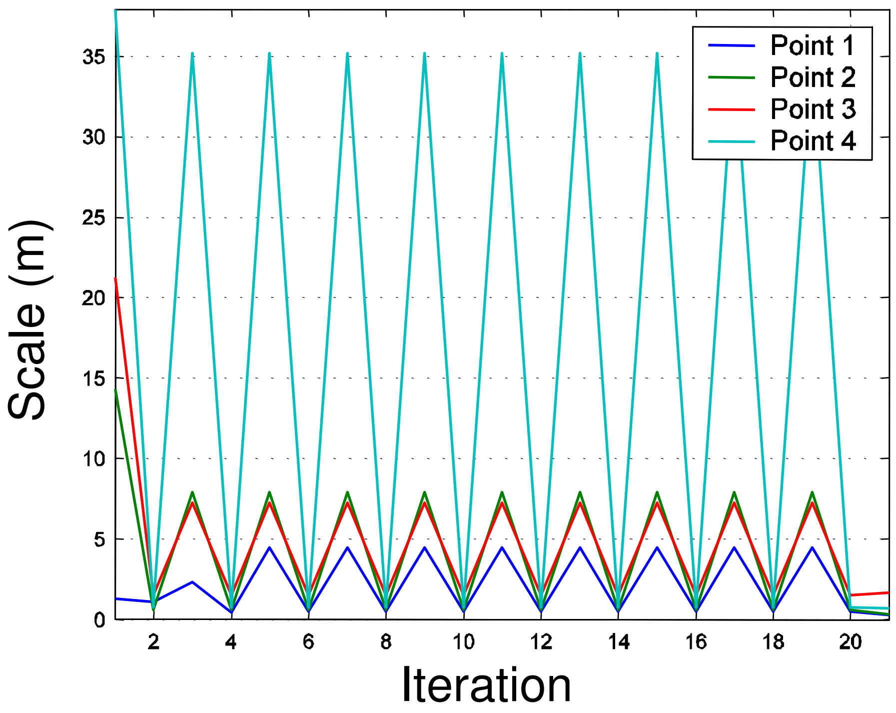
- Initial value of $k=k^{(i)}$ nearest neighbors
- Iterative procedure
  - Estimate curvature $\kappa^{(i)}$ and density $\rho^{(i)}$
  - Compute $r^{(i+1)}$
  - k_{computed} is number of points in neighborhood of size $r^{(i+1)}$
  - Dampening on k:

$$k^{(i+1)} = \gamma k_{\text{computed}} + (1-\gamma)k^{(i)}$$

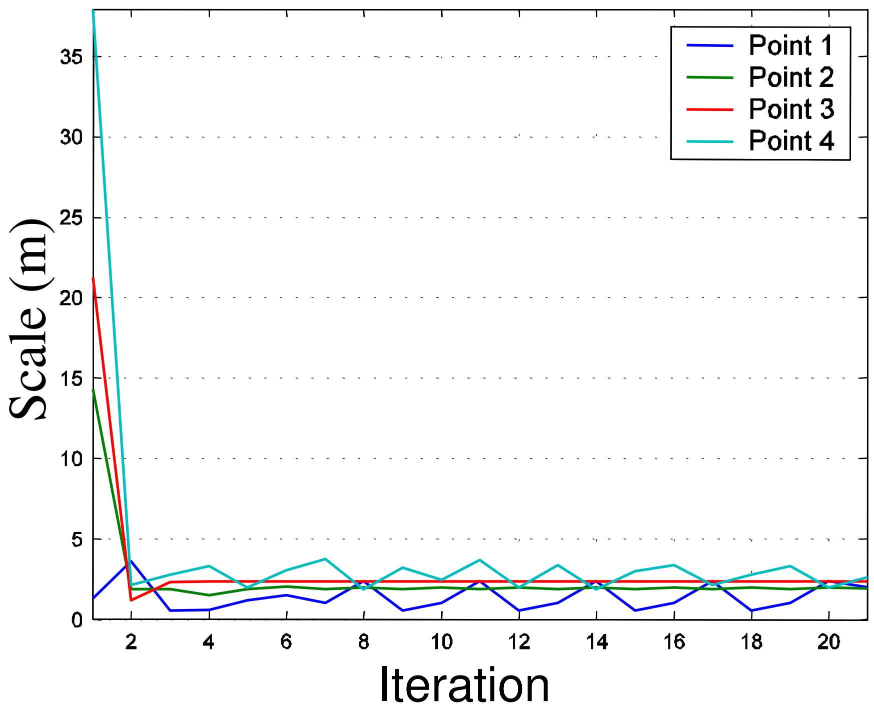
γ
Dampening factor

Effect of dampening on convergence

Original method (no dampening)



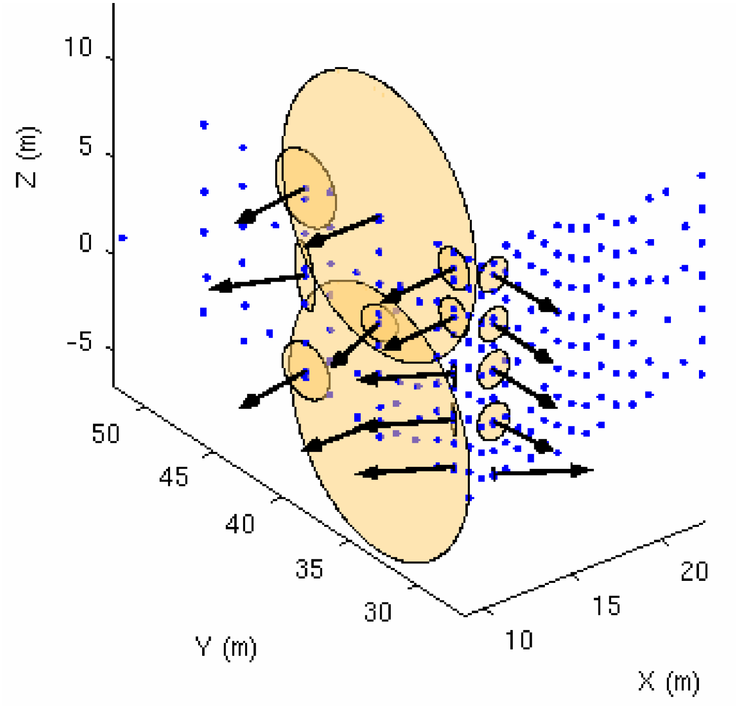
With dampening



Effect of dampening on normal estimation

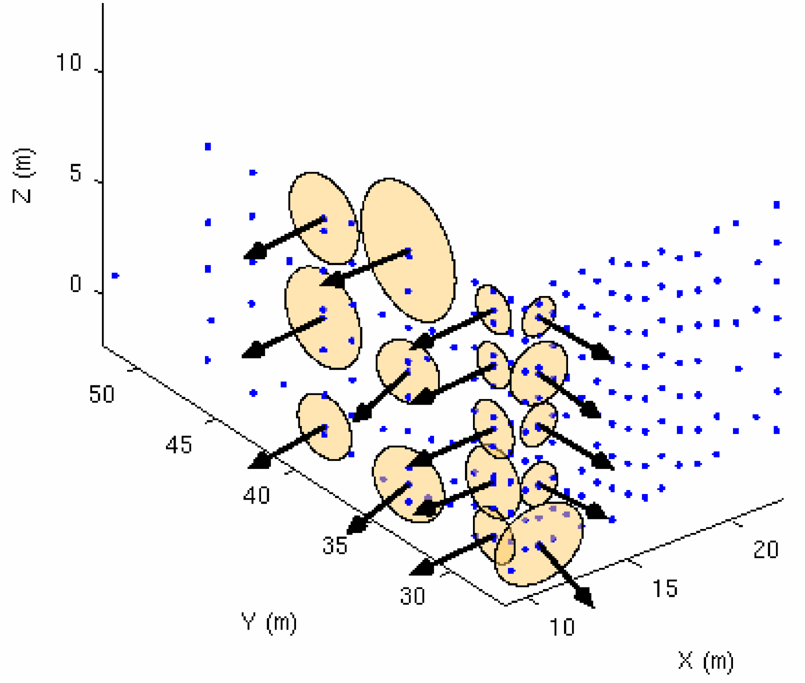
Original method (no dampening)

Avg. error = 22 deg.

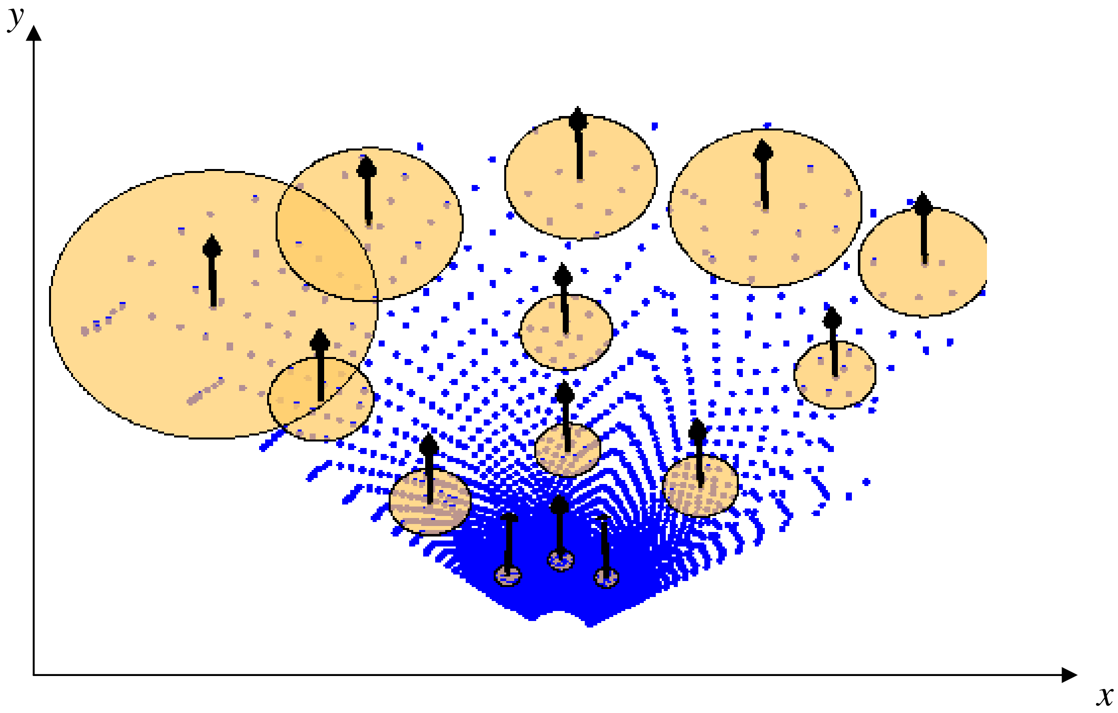


With dampening

Avg. error = 12 deg.



Variation of density

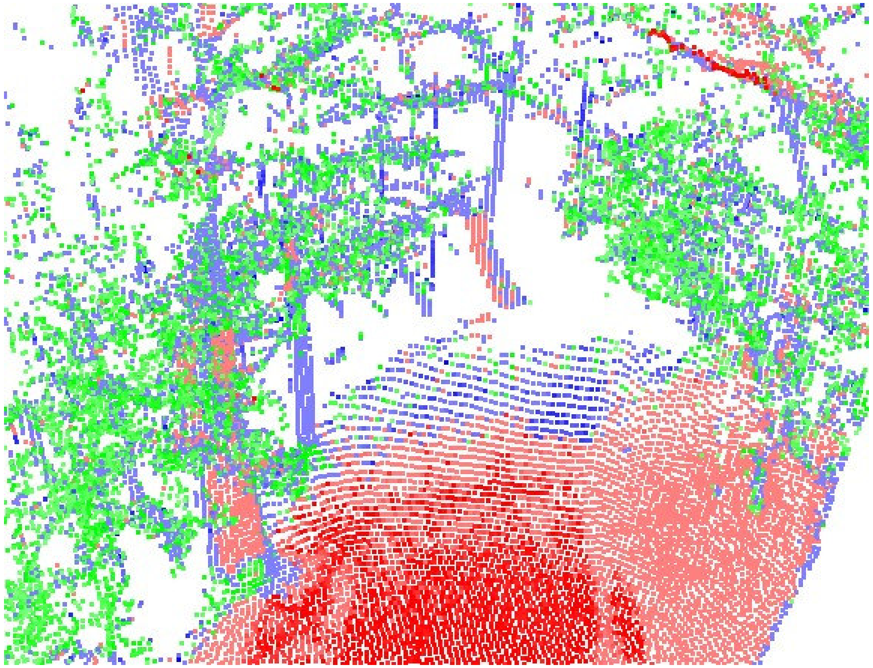


- Data subsampled for clarity
- Normals estimated from support region
- Scale determined by the algorithm

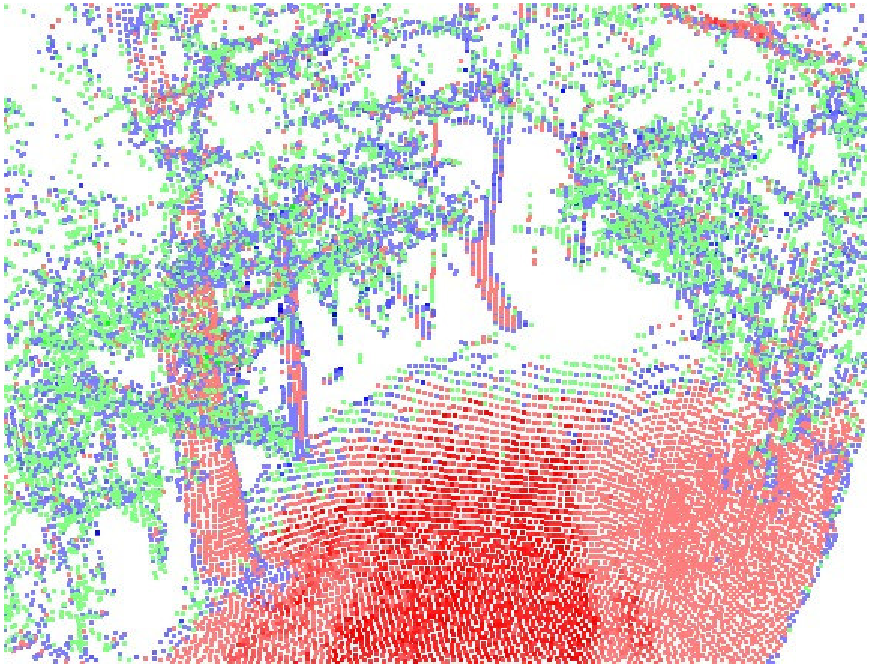
Classification experiments

- SICK scanner

Fixed scale (0.4 m)



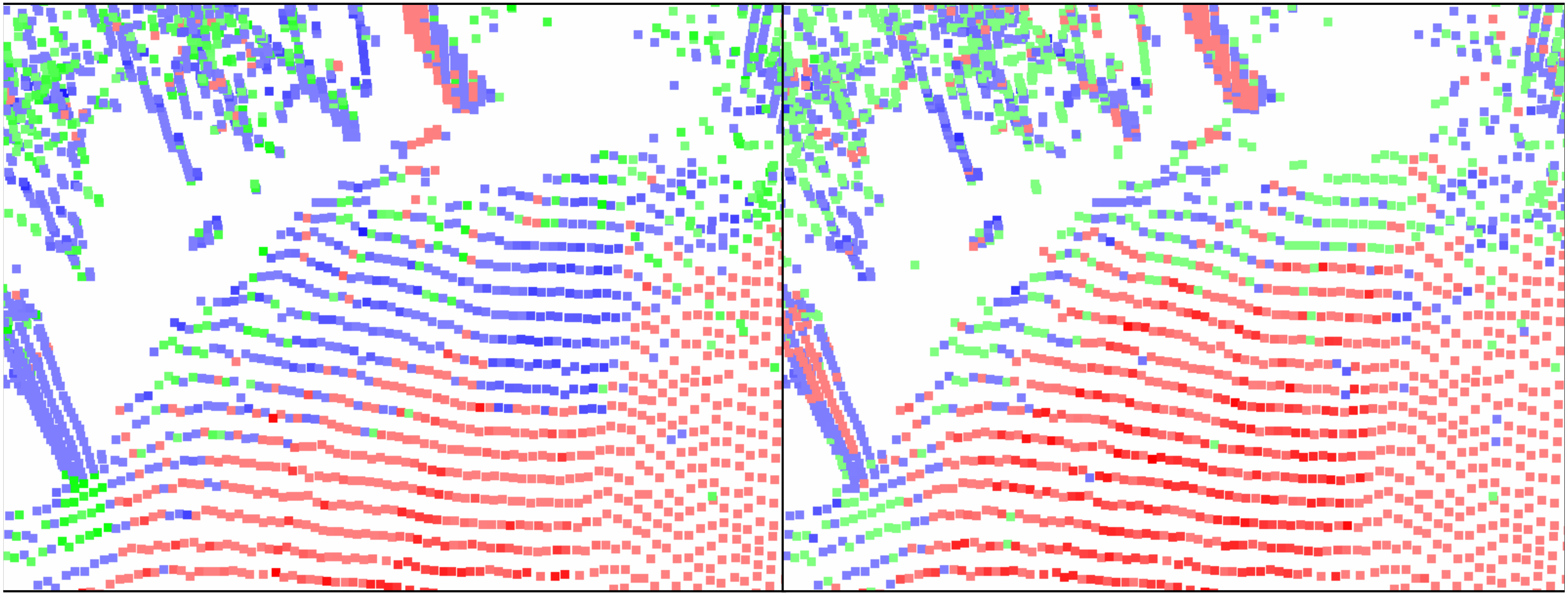
Variable scale at each point



- 0.4m best fixed scale, determined experimentally
- Improvement of 30% for previously misclassified points

Classification experiments

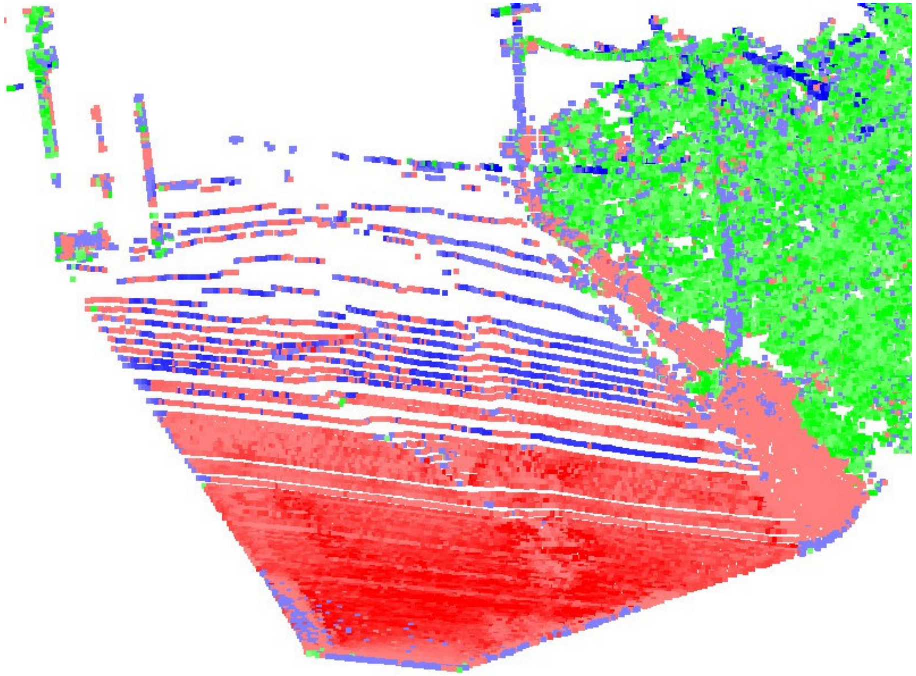
- SICK scanner



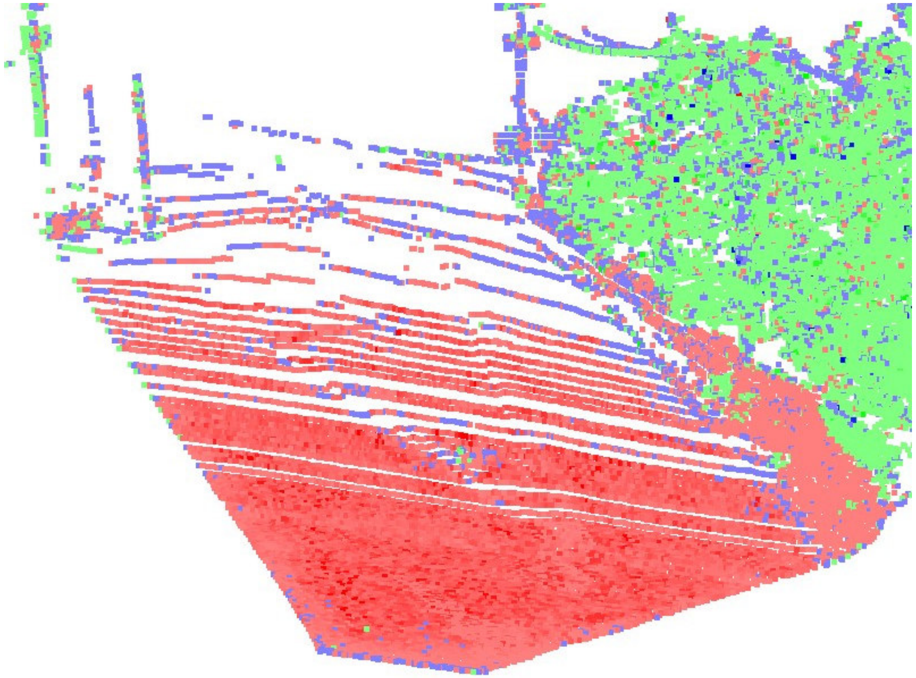
Classification experiments

- RIEGL scanner

Fixed scale (0.4 m)



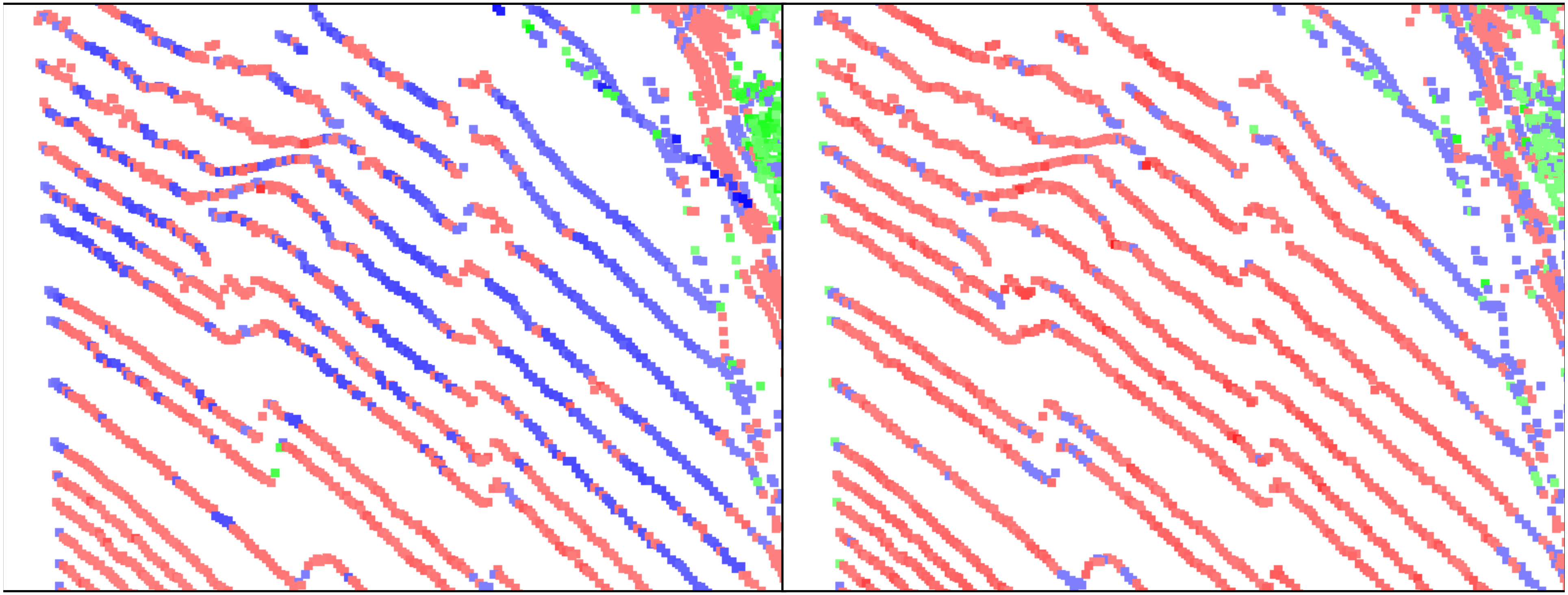
Variable scale at each point



Classification experiments

- RIEGL scanner

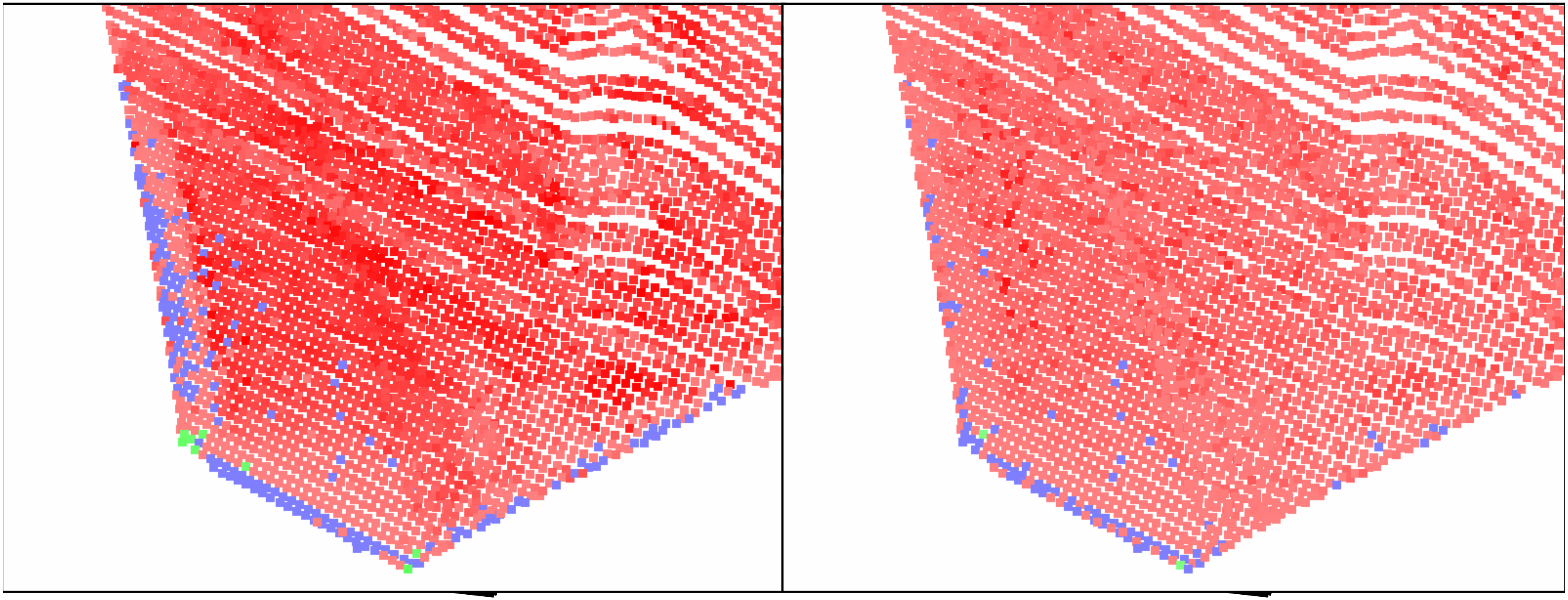
Fixed scale (0.4 m) | Variable scale at each point



Classification experiments

- RIEGL scanner

Fixed scale (0.4 m) | Variable scale at each point



Scale selection – summary

- Problem
  - Optimal scale to best estimate normals
- Approach
  - Use existing approach [Mitra05]
  - Hypothesis

Optimal scale for **geometry** → Good feature for **classification**

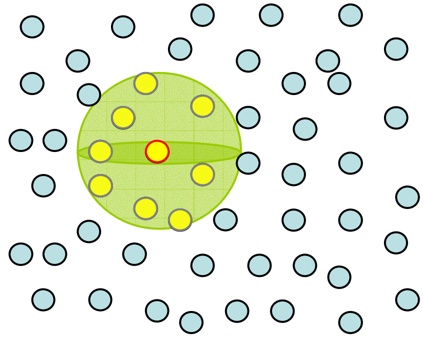
  - Initial experiments show 30% improvement over previously misclassified points
- Future work
  - Different method (more stable)
    - See upcoming 3DPVT paper for linear structures

J.-F. Lalonde, R. Unnikrishnan, N. Vandapel and M. Hebert, "Scale Selection for Classification of Points-Sampled 3-D Surfaces", *Fifth International Conference on 3-D Digital Imaging and Modeling (3DIM)*, 2005.

R. Unnikrishnan, J.-F. Lalonde, N. Vandapel and M. Hebert, "Scale Selection for the Analysis of Point-Sampled Curve", accepted for publication at the *International Symposium on 3-D Data Processing, Visualization and Transmission (3DPVT)*, 2006

Summary

- Improve 3-D signal processing techniques
  - Rely on local computations
- Speed of execution
  - Efficient data structure
- Accuracy
  - 3-D scale selection



- Future work
  - Improve speed for scale
  - Combine 2 techniques

Thank you!

- Any questions?

References

- [Faugeras93] O. Faugeras et al. Real-time correlation-based stereo : algorithm, implementations and applications. Technical Report RR-2013, INRIA, 1993.
- [Gray04] A. Gray and A. Moore. Data structures for fast statistics. Tutorial presented at the International Conference on Machine Learning, 2004.
- [Hebert03] M. Hebert and N. Vandapel. Terrain Classification Techniques from LADAR data for Autonomous Navigation, *Proc. of the Collaborative Technology Alliance Conference*, 2003
- [Lacaze02] A. Lacaze, K. Murphy, and M. DelGiorno. Autonomous mobility for the demo III experimental unmanned vehicles. In *Proc. of the AUVSI Conference*, 2002.
- [Lalonde05] J.-F. Lalonde, R. Unnikrishnan, N. Vandapel and M. Hebert, "Scale Selection for Classification of Points-Sampled 3-D Surfaces", *Fifth International Conference on 3-D Digital Imaging and Modeling (3DIM)*, 2005.
- [Lindeberg90] T. Lindeberg. Scale-space for discrete signals. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(3), 1990.
- [Liu04] T. Liu, A. Moore, A. Gray, and K. Yang. An investigation of practical approximate nearest neighbor algorithms. In *Neural Information Processing Systems*, 2004.
- [Mitra05] N. Mitra, A. Nguyen and L. Guibas, Estimating surface normals in noisy point cloud data. *Intl. Journal of Computational Geometry and Applications*, 2005.
- [Pauly03] M. Pauly, R. Keiser, and M. Gross. Multi-scale feature extraction on point-sampled surfaces. In *Eurographics*, 2003.
- [Samet89] H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, 1989.
- [Tang04a] C. Tang, G. Medioni, P. Mordohai, and W. Tong. First order augmentations to tensor voting for boundary inference and multiscale analysis in 3-d. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(5), 2004.
- [Tang04b] F. Tang, M. Adams, J. Ibanez-Guzman, and W. Wijesoma. Pose invariant, robust feature extraction from range data with a modified scale space approach. In *IEEE Intl. Conf. on Robotics and Automation*, 2004.
- [Unnikrishnan06] R. Unnikrishnan, J.-F. Lalonde, N. Vandapel and M. Hebert, "Scale Selection for the Analysis of Point-Sampled Curve", accepted for publication at the *International Symposium on 3-D Data Processing, Visualization and Transmission (3DPVT)*, 2006
- [Vandapel04] N. Vandapel, D. Huber, A. Kapuria, and M. Hebert. Natural Terrain Classification using 3-D Ladar Data. In *IEEE International Conference on Robotics and Automation*, April 2004

Additional slides…

Goal

Improve perception capabilities of outdoor ground mobile robots

- “Bare ground” environments
  - Dense obstacles (e.g. rocks)
  - On the ground
- A 2-D representation is sufficient
  - 2-D-½ (with elevation)
  - Convolve vehicle model
- Towards more challenging environments
  - Vegetation (porous obstacles)
  - Thin structures (branches)
  - Overhanging obstacles
- Need a 3-D representation

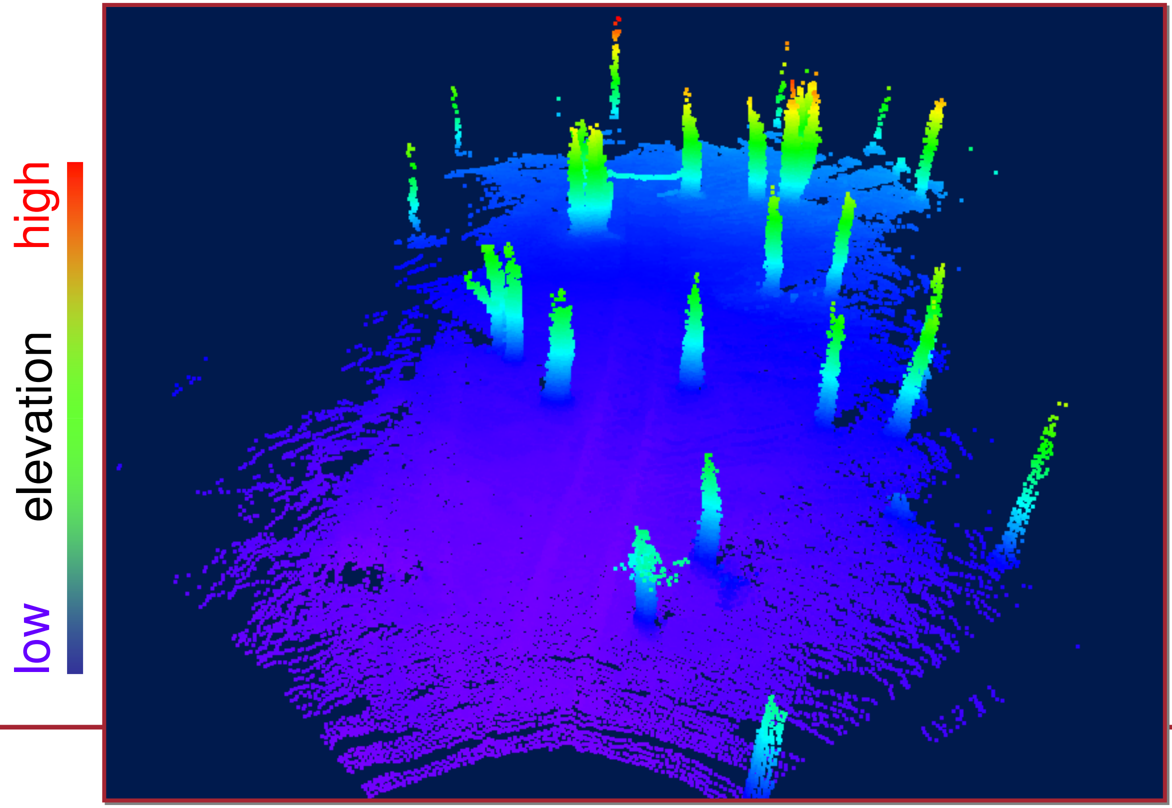
Overview – Robot

- GDRS eXperimental Unmanned Vehicle
- GDRS Mobility LADAR
  - Time-of-flight
- Mounted on turret
- 100,000 3-D points per second

Robot

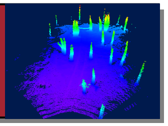
Overview – Raw 3-D points

- 3-D point cloud
  - Points are co-registered *wrt* global ref. frame
    - From robot's IMU
  - Accumulated over time
  - Unorganized



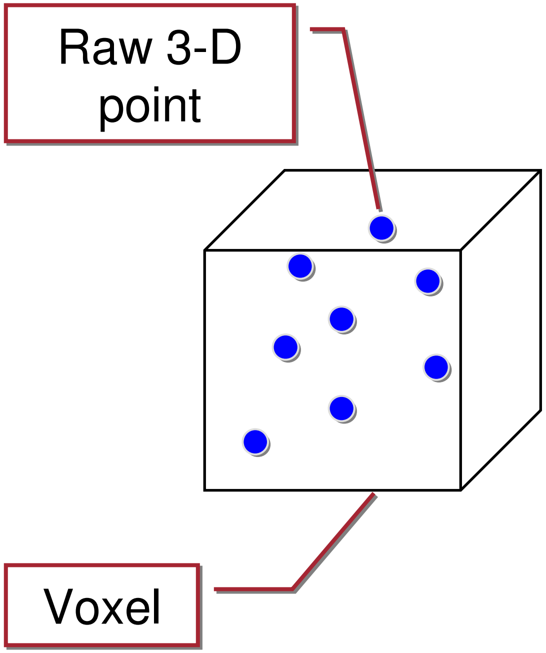
Robot

3-D points



Overview – Voxelization

- Regular grid
  - Basic unit: voxel
- Lossless “compression” scheme
  - Store sufficient statistics for features

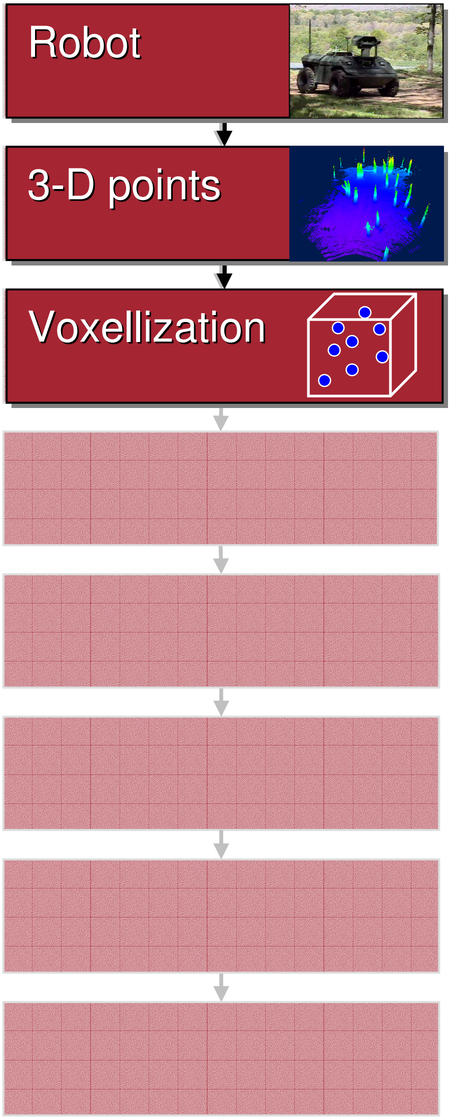


$$\sum_i x_i \sum_i y_i \sum_i z_i$$

$$\sum_i x_i^2 \sum_i y_i^2 \sum_i z_i^2$$

$$\sum_{i,j} x_i y_j \sum_{i,j} x_i z_j \sum_{i,j} y_i z_j$$

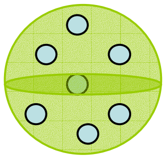
$$n$$



Overview – Feature computation

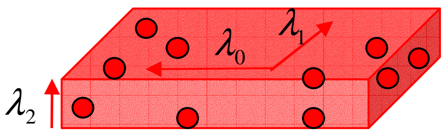
- For each voxel
  - Define local neighborhood
    - Fixed (pre-determined) size
  - Perform range search
    - Loop over all voxels in neighborhood
  - PCA features
    - Eigenvalues of covariance matrix
    - 3 features:

scatter
$\lambda_0 \approx \lambda_1 \approx \lambda_2$



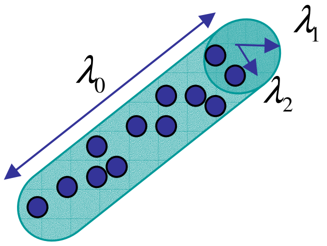
$F_{\text{scatter}} = \lambda_0$

surface
$\lambda_0 \approx \lambda_1 >> \lambda_2$

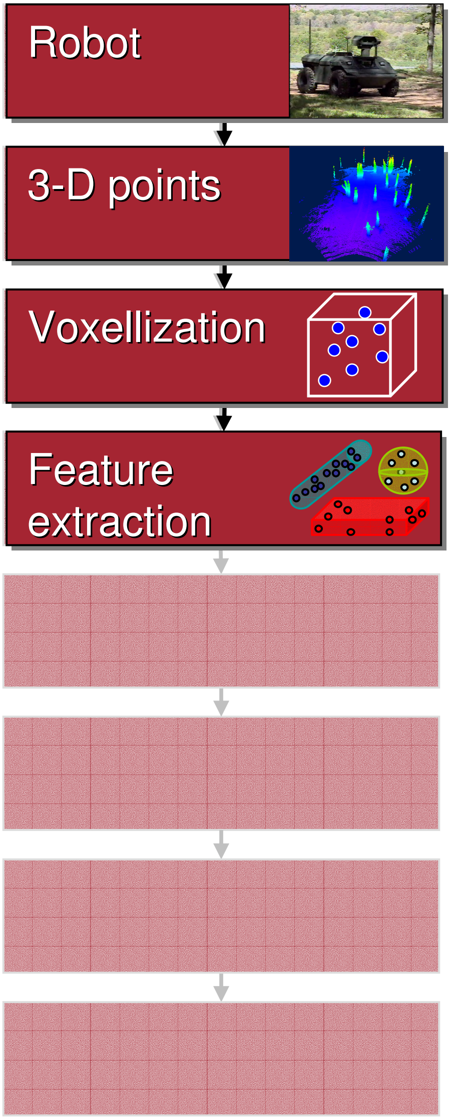


$F_{\text{surface}} = (\lambda_1 - \lambda_2)\cdot \vec{e}_2$

linear
$\lambda_0 >> \lambda_1 \approx \lambda_2$



$F_{\text{linear}} = (\lambda_0 - \lambda_1)\cdot \vec{e}_0$

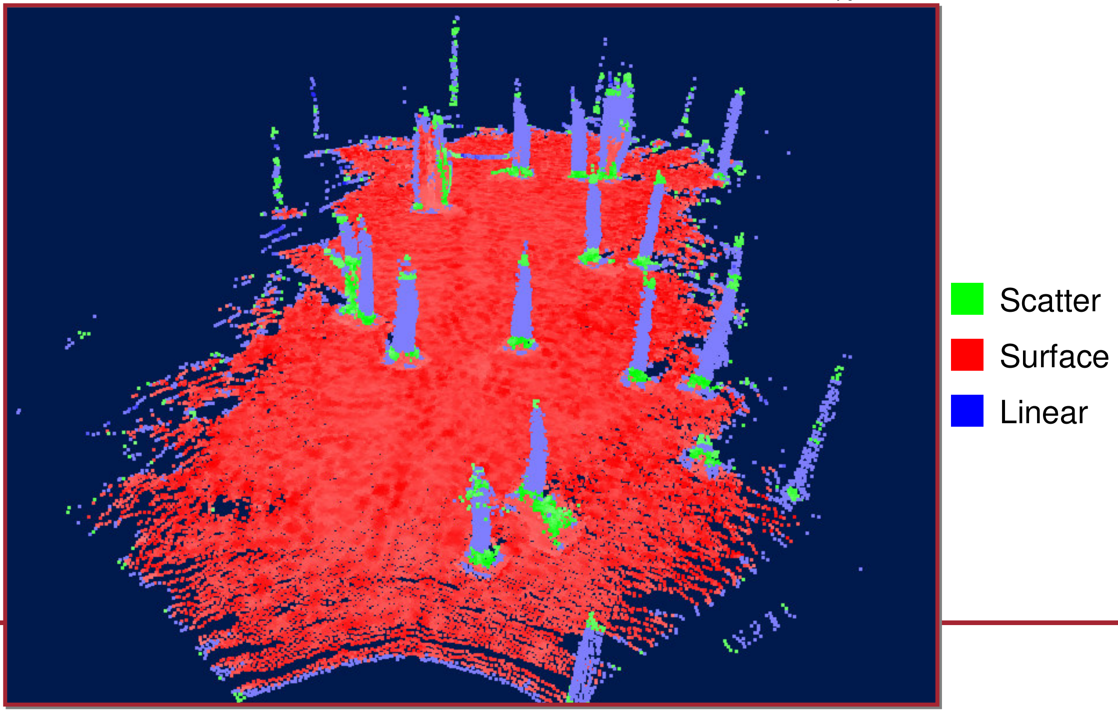


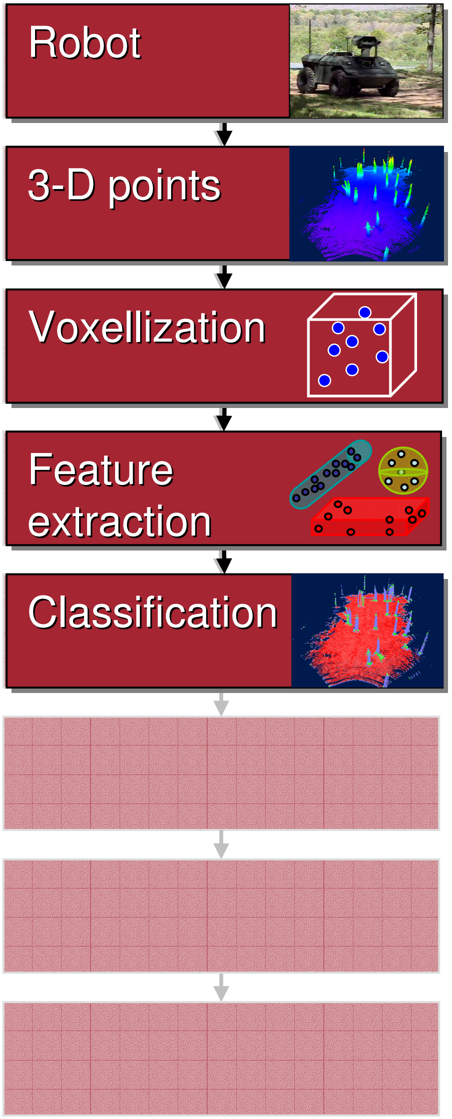
Overview – Classification

- Gaussian Mixture Model
  - 3 gaussians per class (cross-validation)

$$p(F|M^k) = \sum_{i=1\ldots n_g^k} \frac{\omega_i^k}{(2\pi)^{d/2}|\Sigma_i^k|^{1/2}} e^{-\frac{1}{2}(F-\mu_i^k)^T {\Sigma_i^k}^{-1}(F-\mu_i^k)}$$

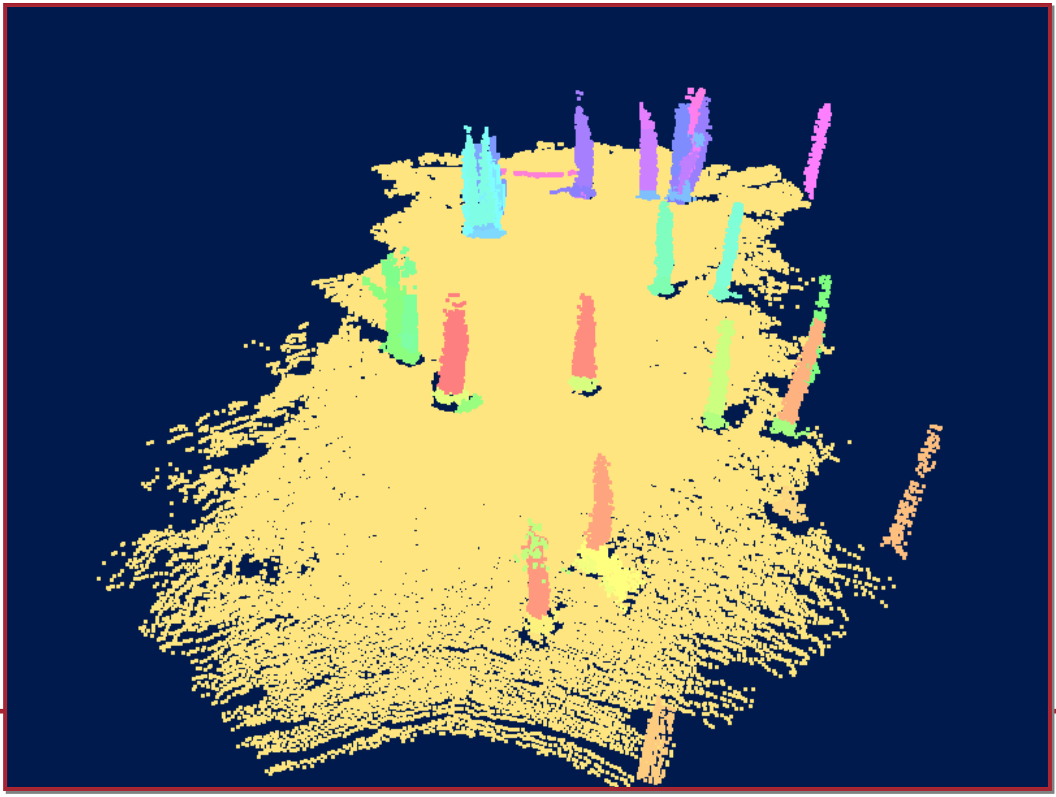
  - Max-likelihood class $k_{max} = \underset{k}{\operatorname{argmax}}\{p(F|M^k)\}$

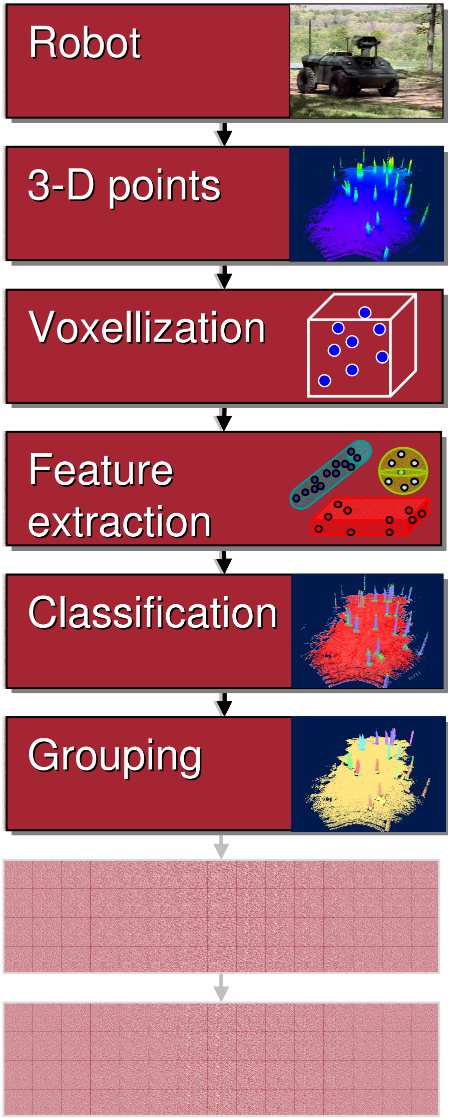




Overview – Grouping

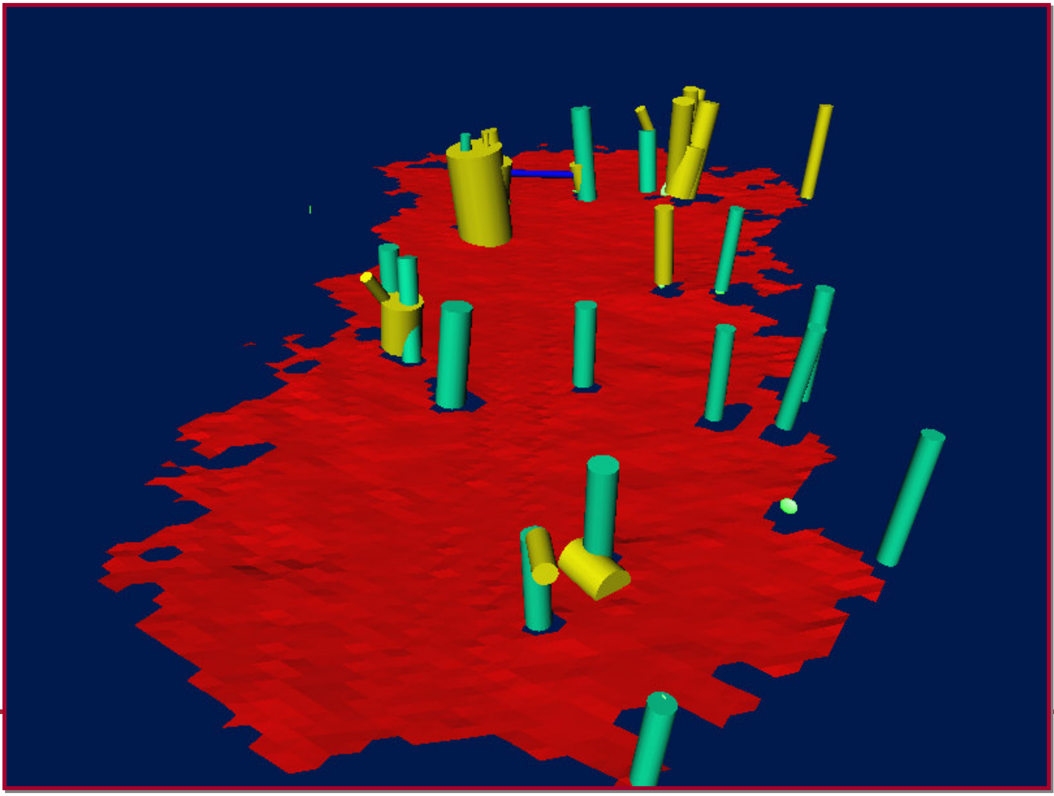
- Connected components algorithm
- Criteria
  - Distance
  - Same class
  - Similar direction

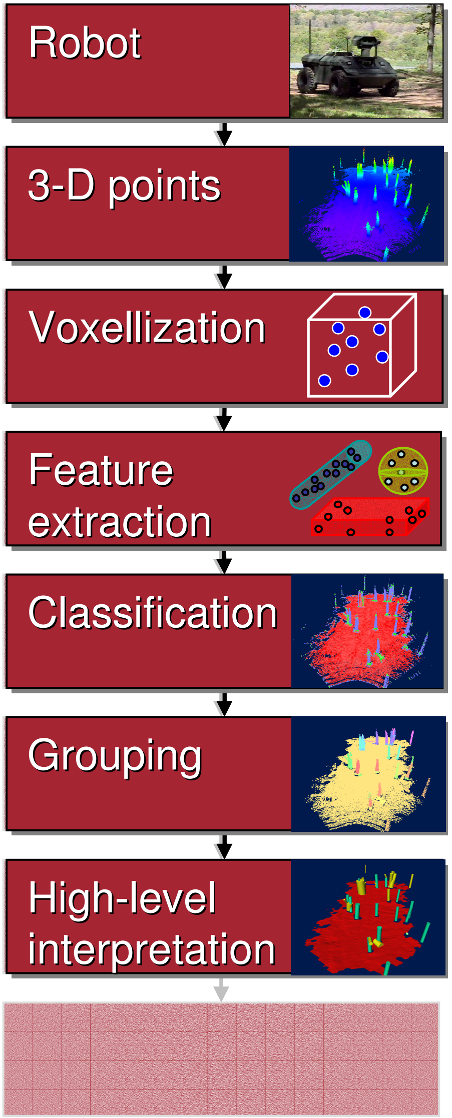




Overview – High-level interpretation

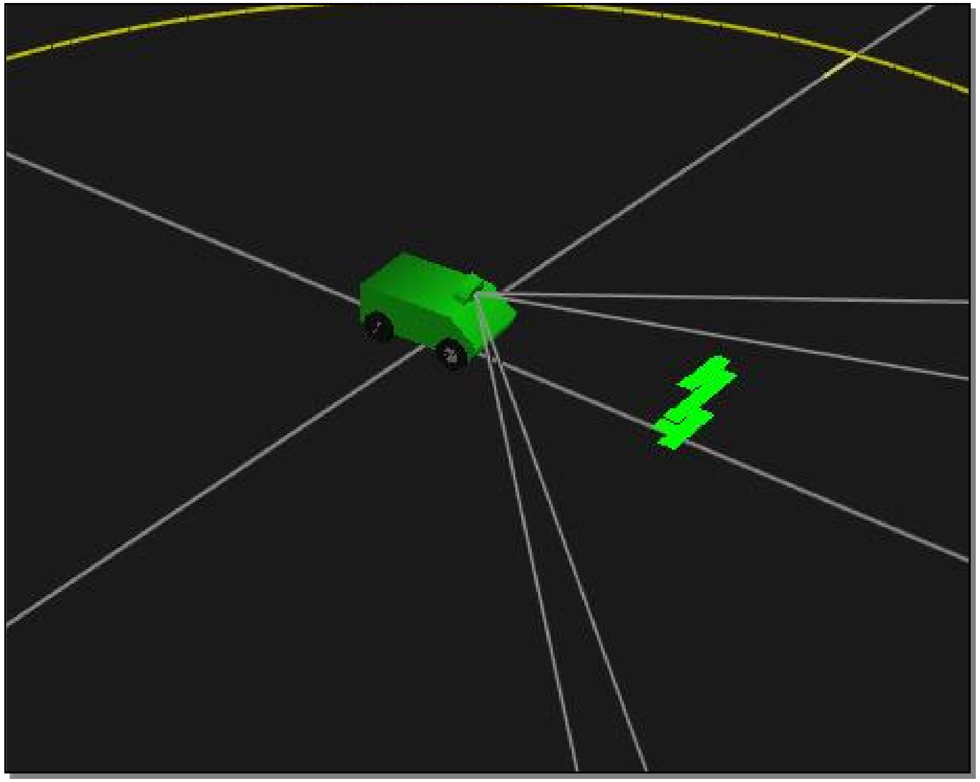
- Object identification
  - Heuristics-based
- Distinguish between similar objects
  - Branches vs wires
  - Tree trunks vs branches



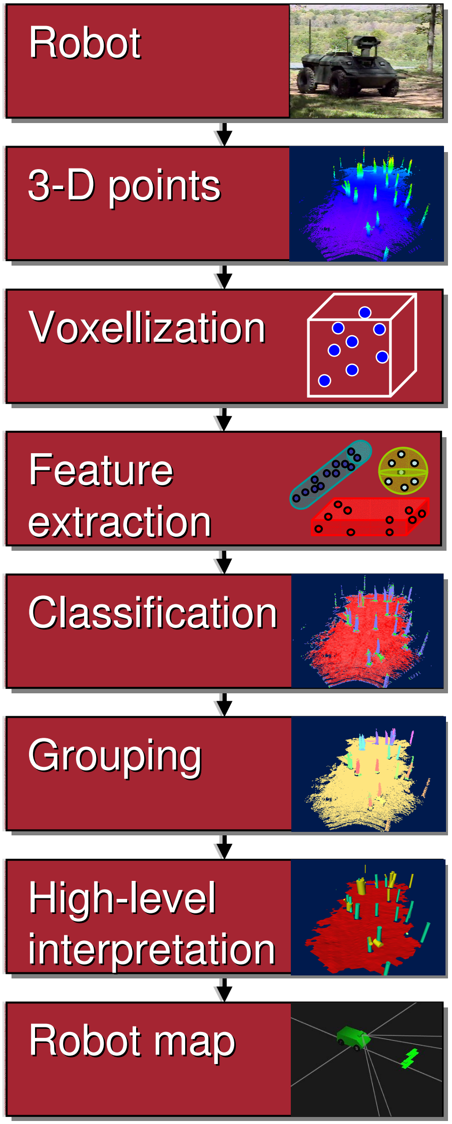


Overview – Robot obstacle map

- Location of obstacles are sent to XUV
- Integration in obstacle map for planning

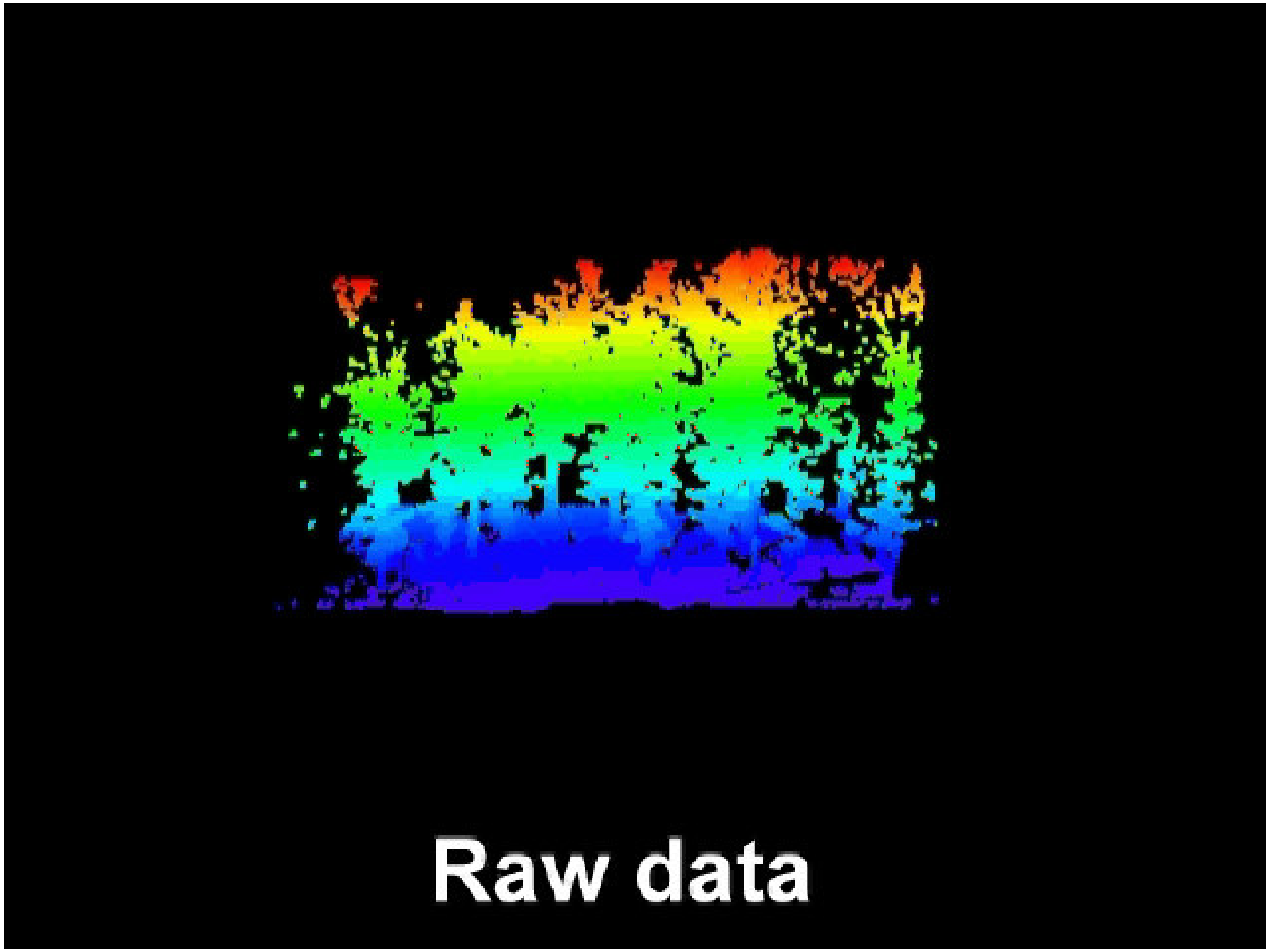


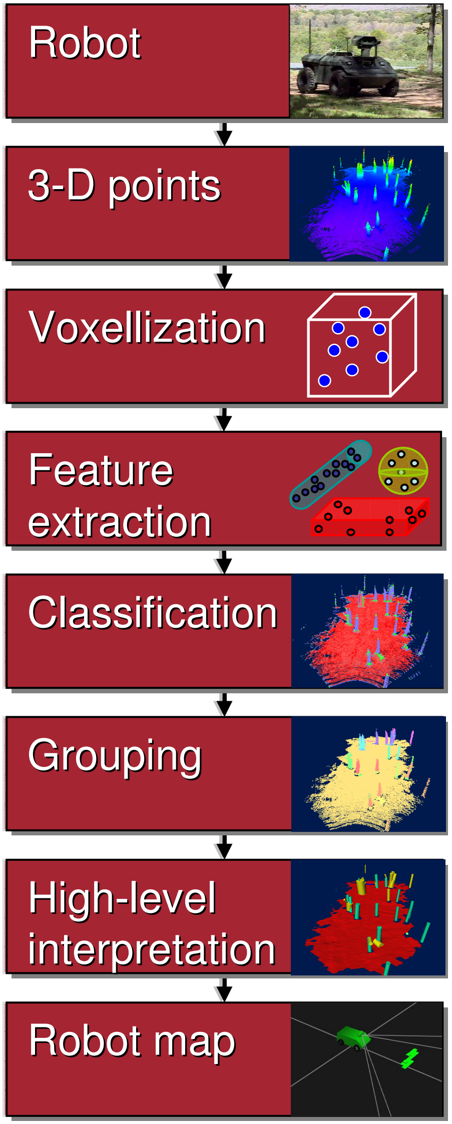
Snapshot of GDRS “IDISP” interface



Overview – examples

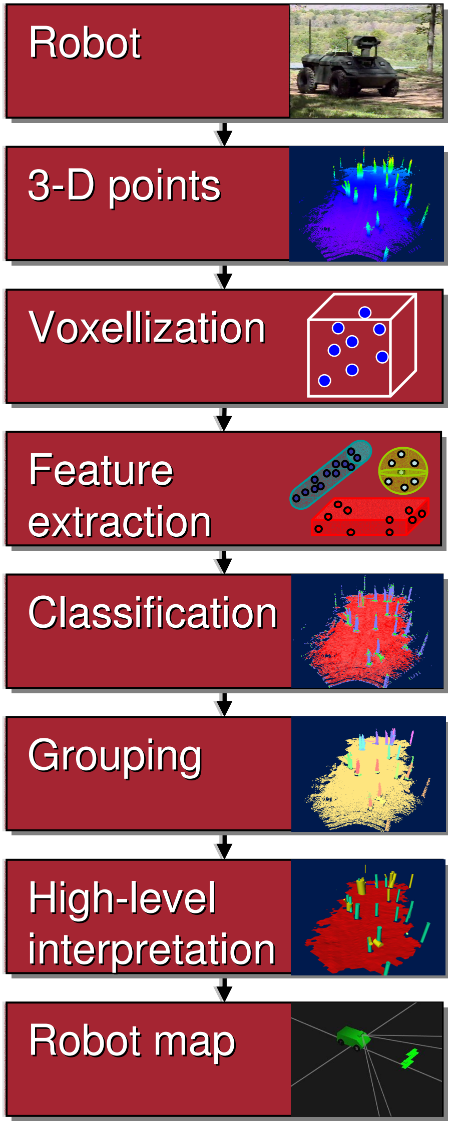
- Automatic tree trunk diameter estimation





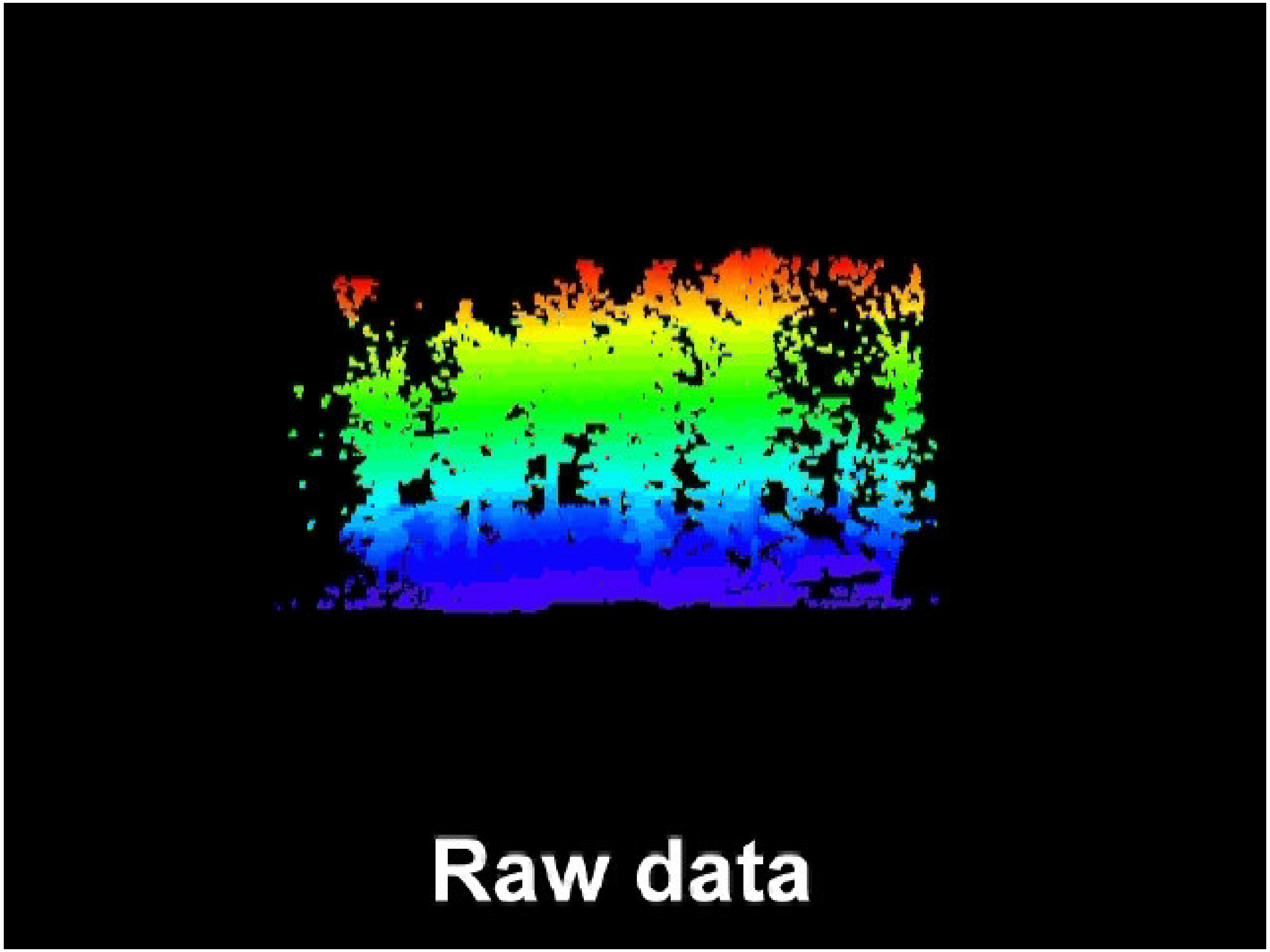
Overview – examples

- Wire detection



Perception for robot mobility (contd.)

- Automatic tree trunk diameter estimation



Problems: Feature computation

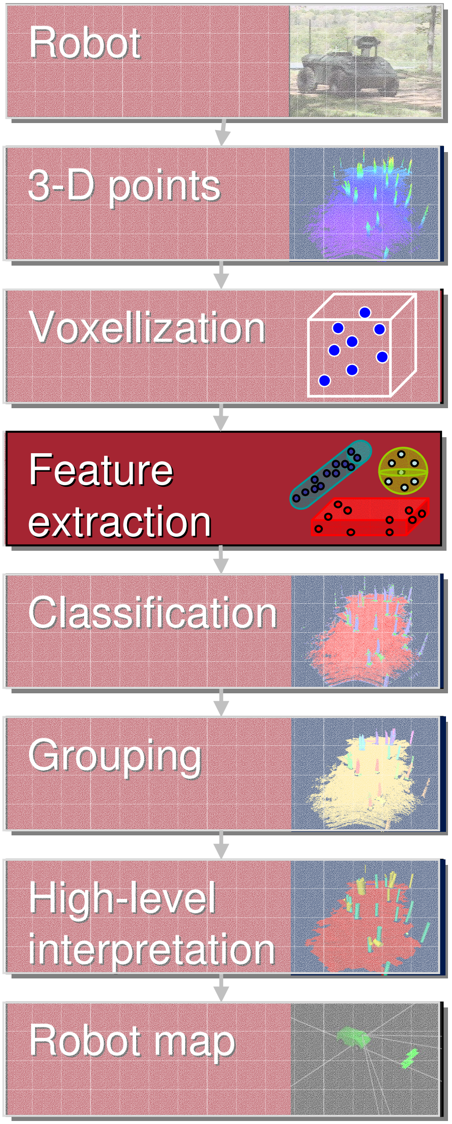
- For each voxel
  - Define local neighborhood
    - Fixed (pre-determined) size
  - Perform range search
    - Loop over all voxels in neighborhood
  - PCA features
    - ...lues of cova...
    - ...es

How to choose the size (scale)?

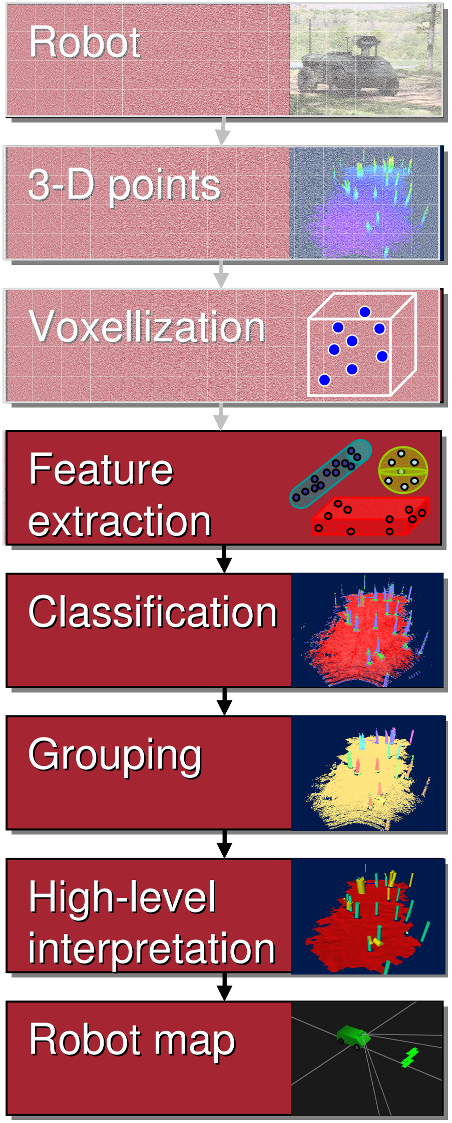
Very expensive operation!

Proposed solutions

- Algorithmic: Efficient data structure
- Analytic: Automatic scale selection



Summary



Proposed solutions

- Algorithmic: Efficient data structure
- Analytic: Automatic scale selection

Experiments – dynamic data

- Batch timing definition not suitable
  - Frame rate
  - Vehicle speed
- New definition
  - Tied to obstacle detection
  - Time between voxel creation and classification
  - Cumulative histograms