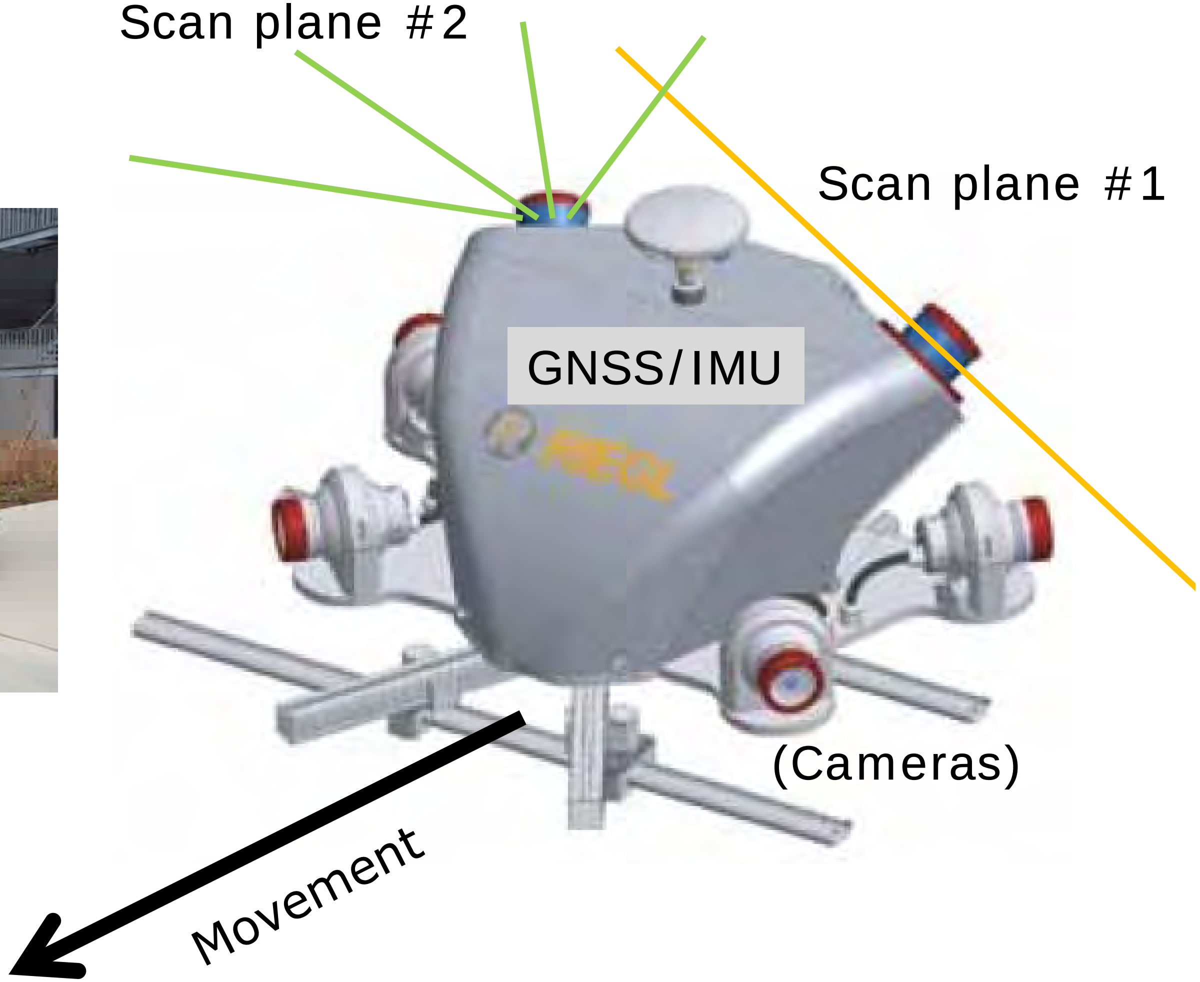
Institute of Cartography and Geoinformatics | Leibniz Universität Hannover

Localization and Mapping: Adjustment and Interpretation of LIDAR Data

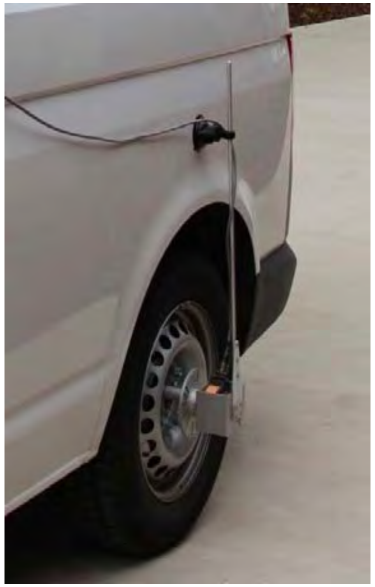
Claus.Brenner@ikg.uni-hannover.de

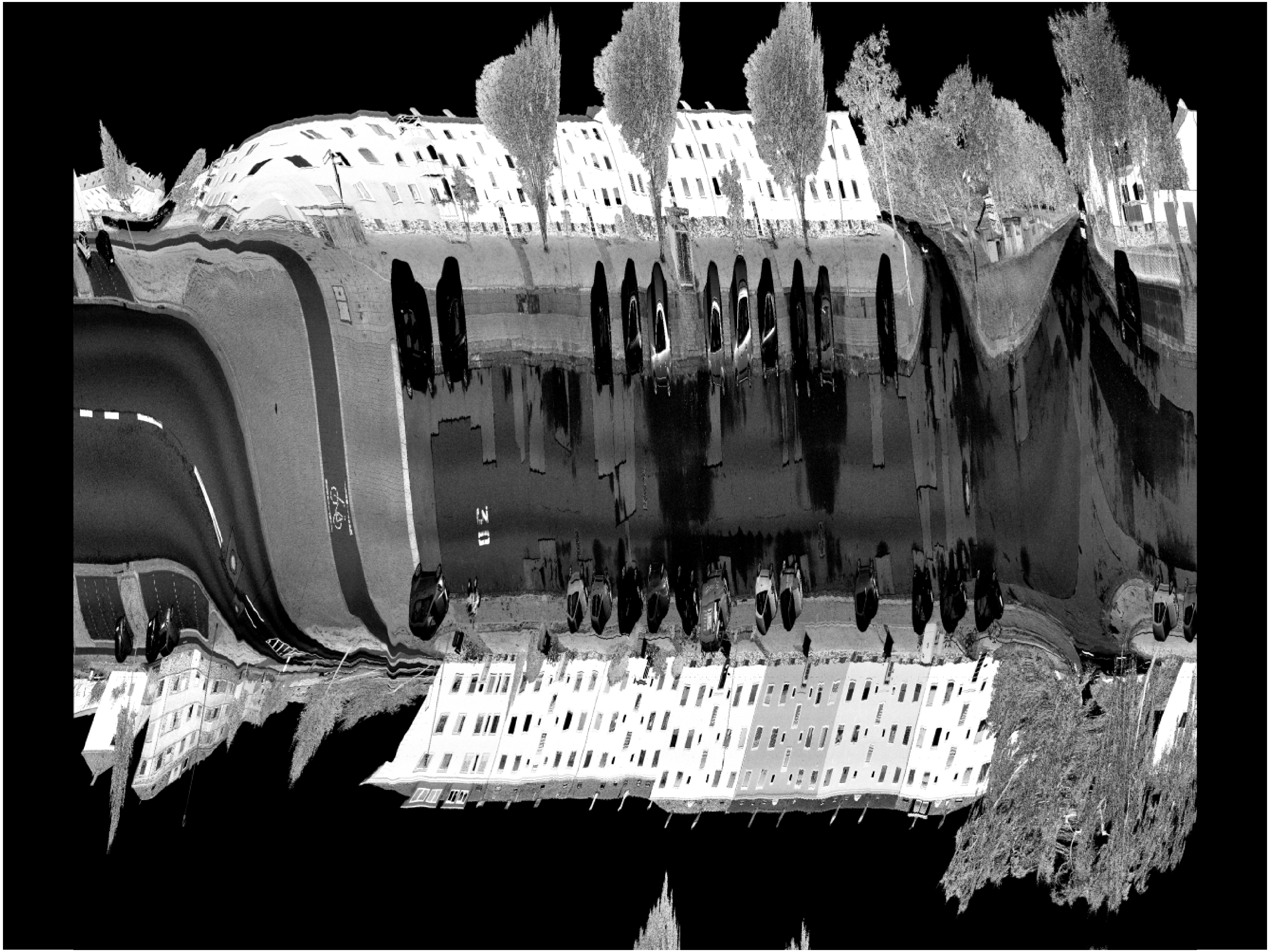
Leibniz Universität Hannover

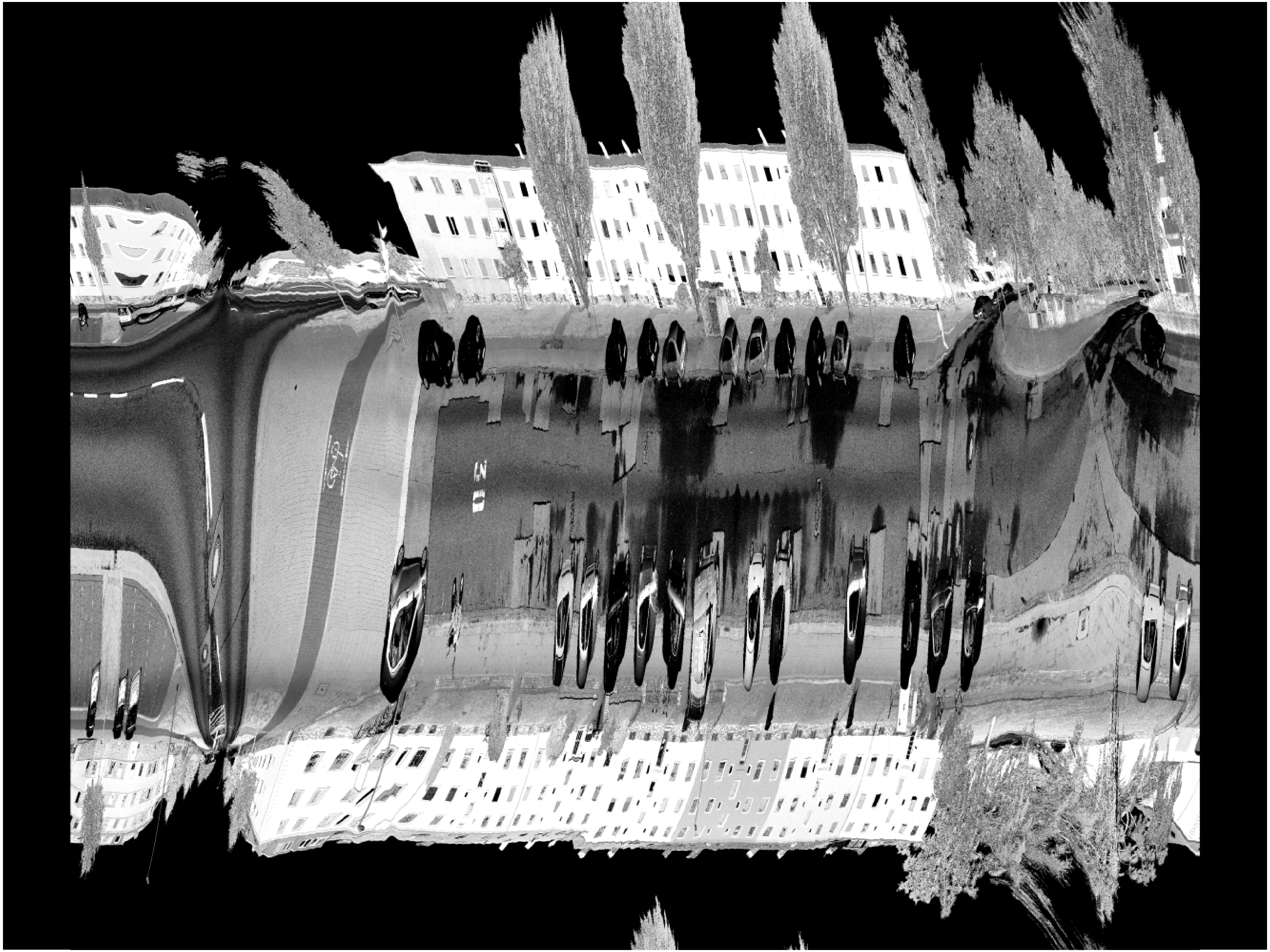
LiDAR Mobile Mapping: Riegl VMX-250



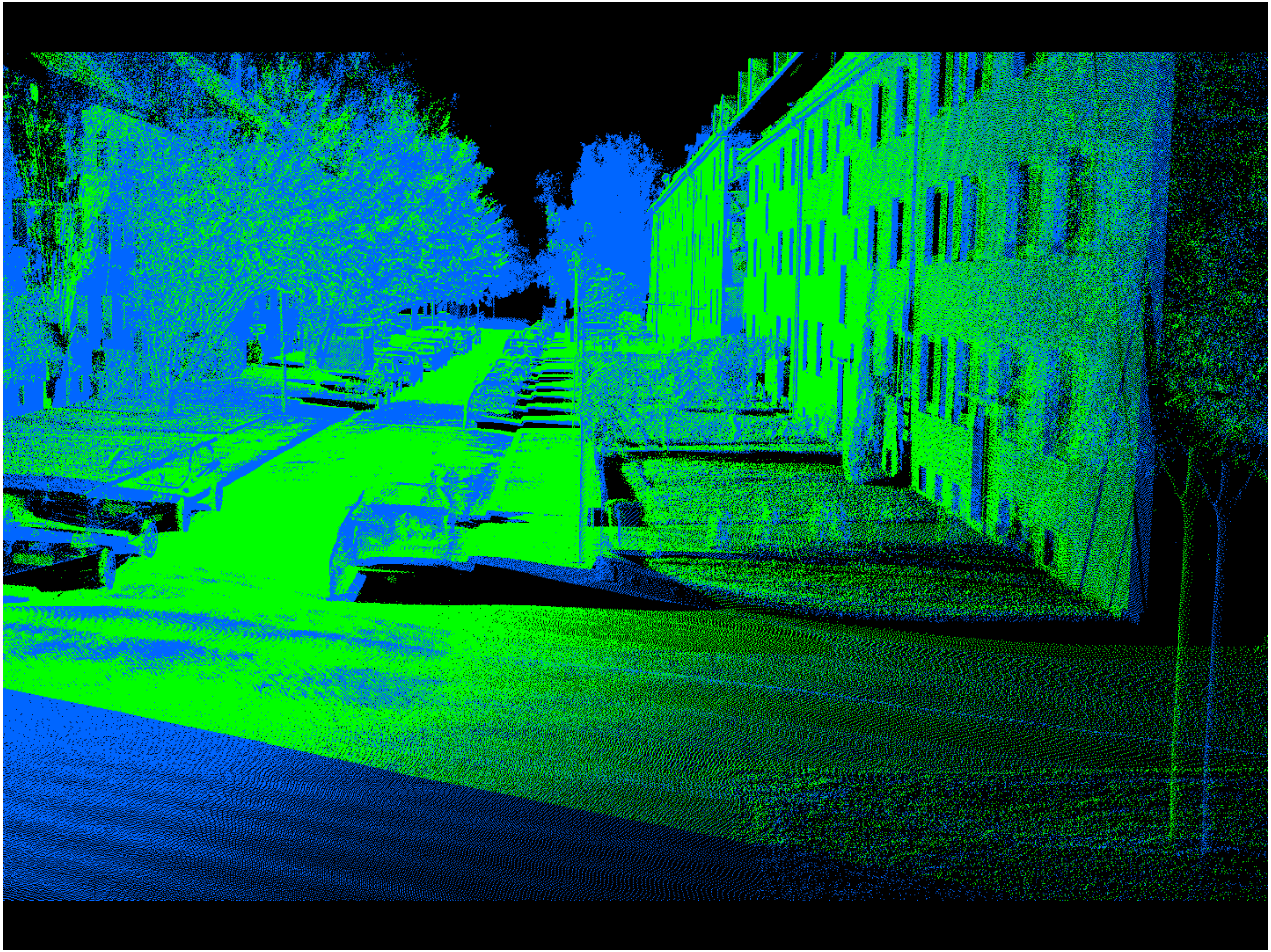
LiDAR Mobile Mapping: Riegl VMX-250

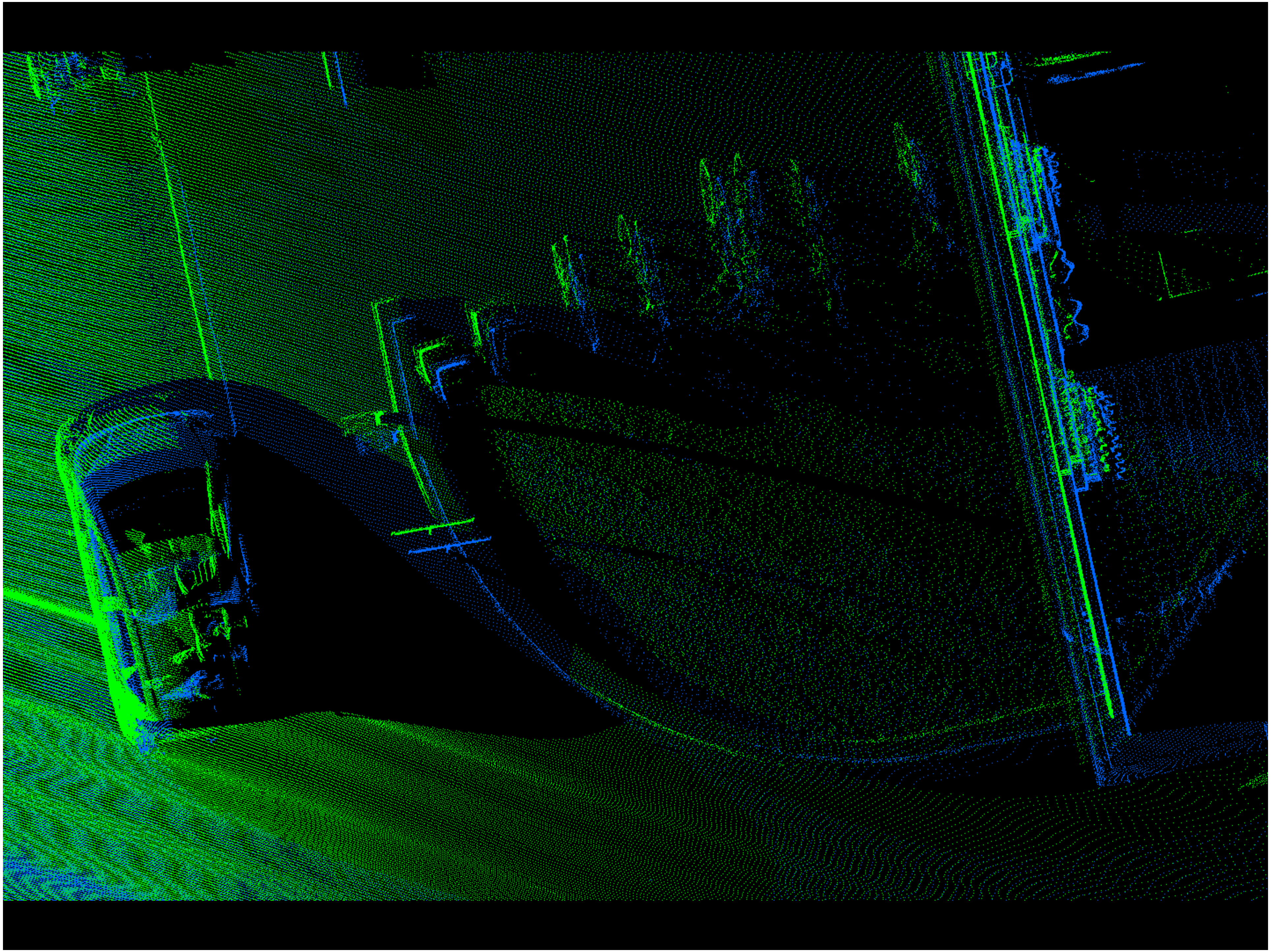




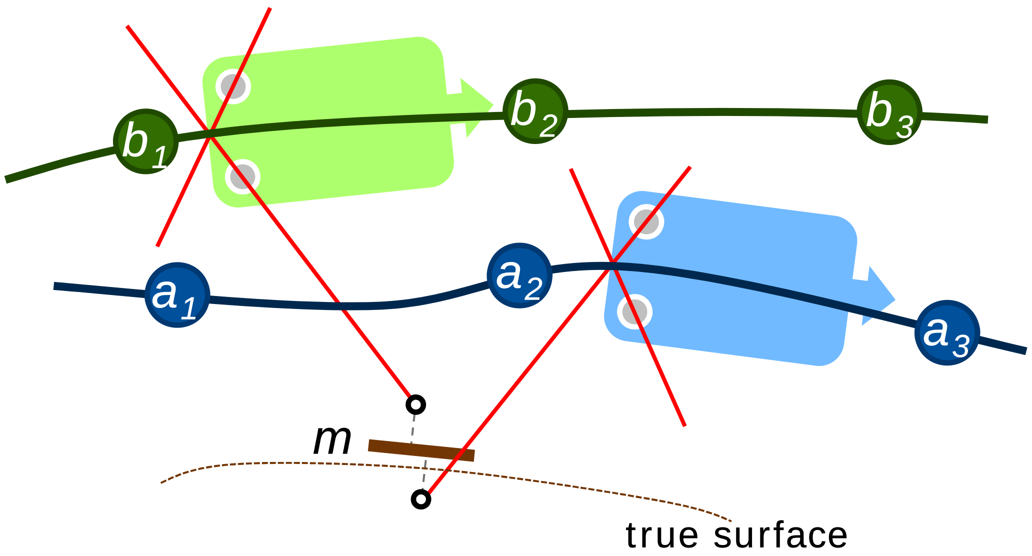


Adjustment of huge datasets



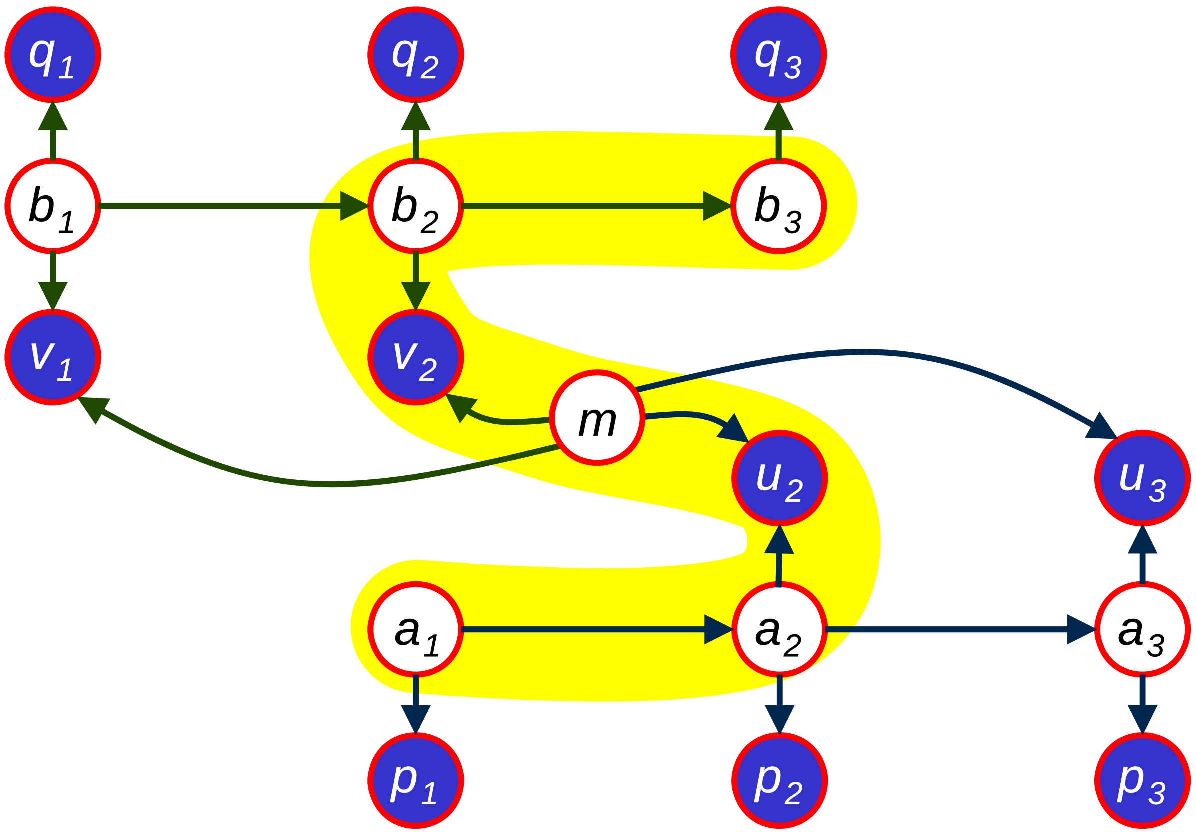


Latent map approach



- Latent map element *m*
- Basic principle of photogrammetric bundle adjustment (tie points)
- Cost: additional unknowns $\propto$ total surface

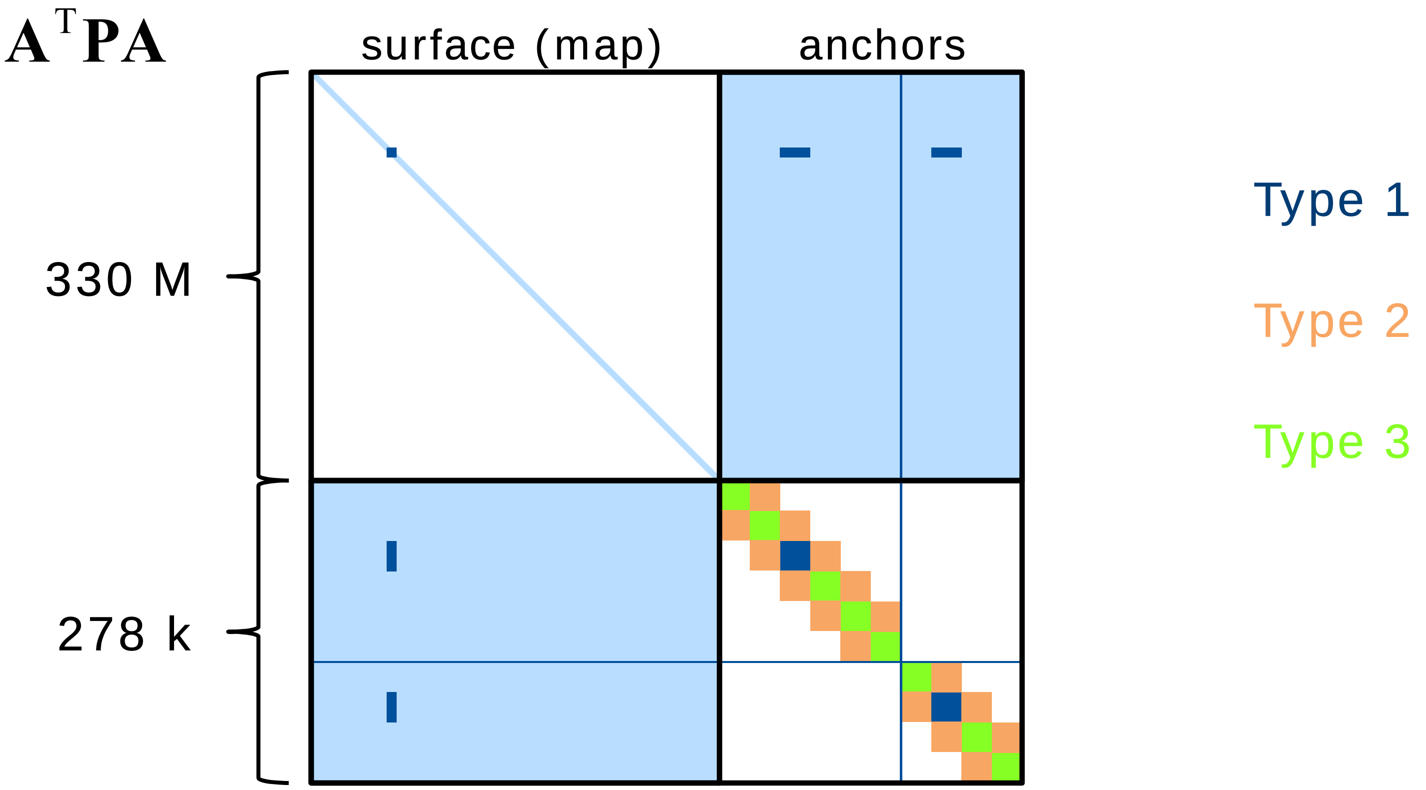
Overall Bayes network



MLE: $$X^* = \underset{a_1, a_2, \ldots b_3, m}{\arg\max}\, P(a_1, a_2, a_3, b_1, b_2, b_3, m) = \underset{A,M}{\arg\max}\, P(A, M)$$

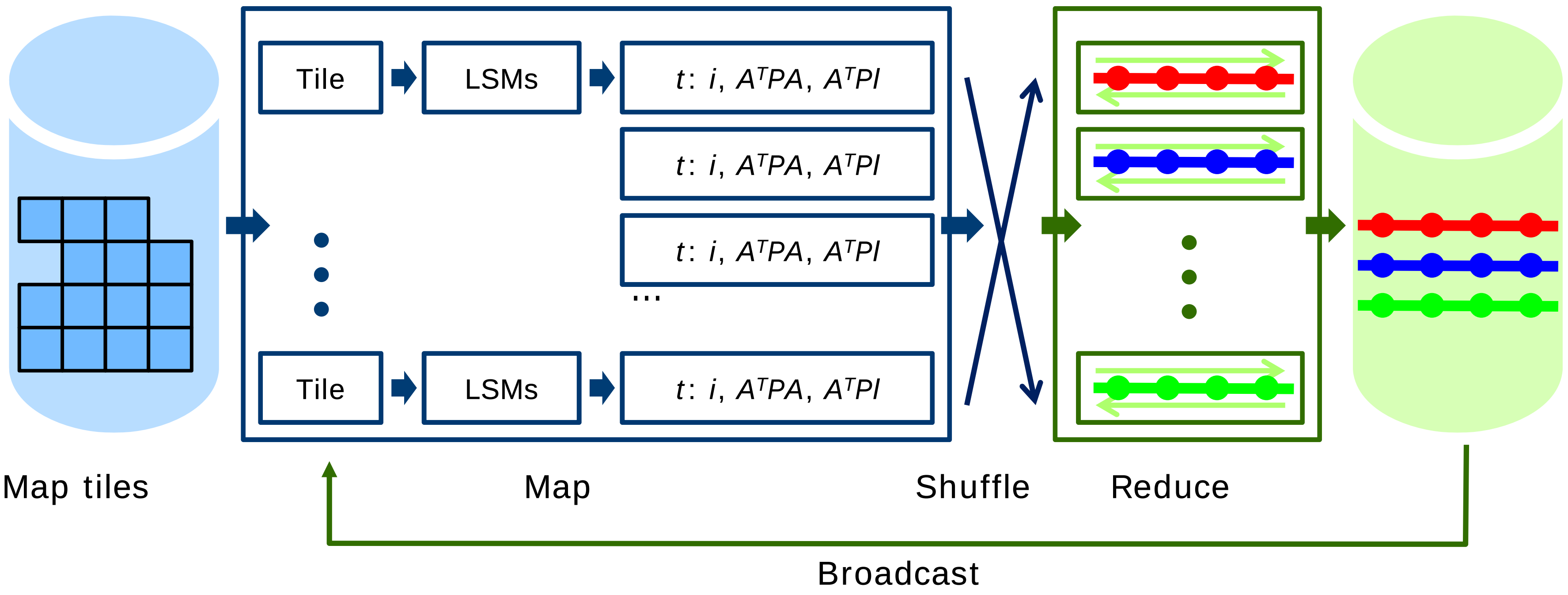
Matrix layout

- Solution: $\mathbf{X}^* = \left(\mathbf{A}^T\mathbf{P}\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{P}\mathbf{l}$

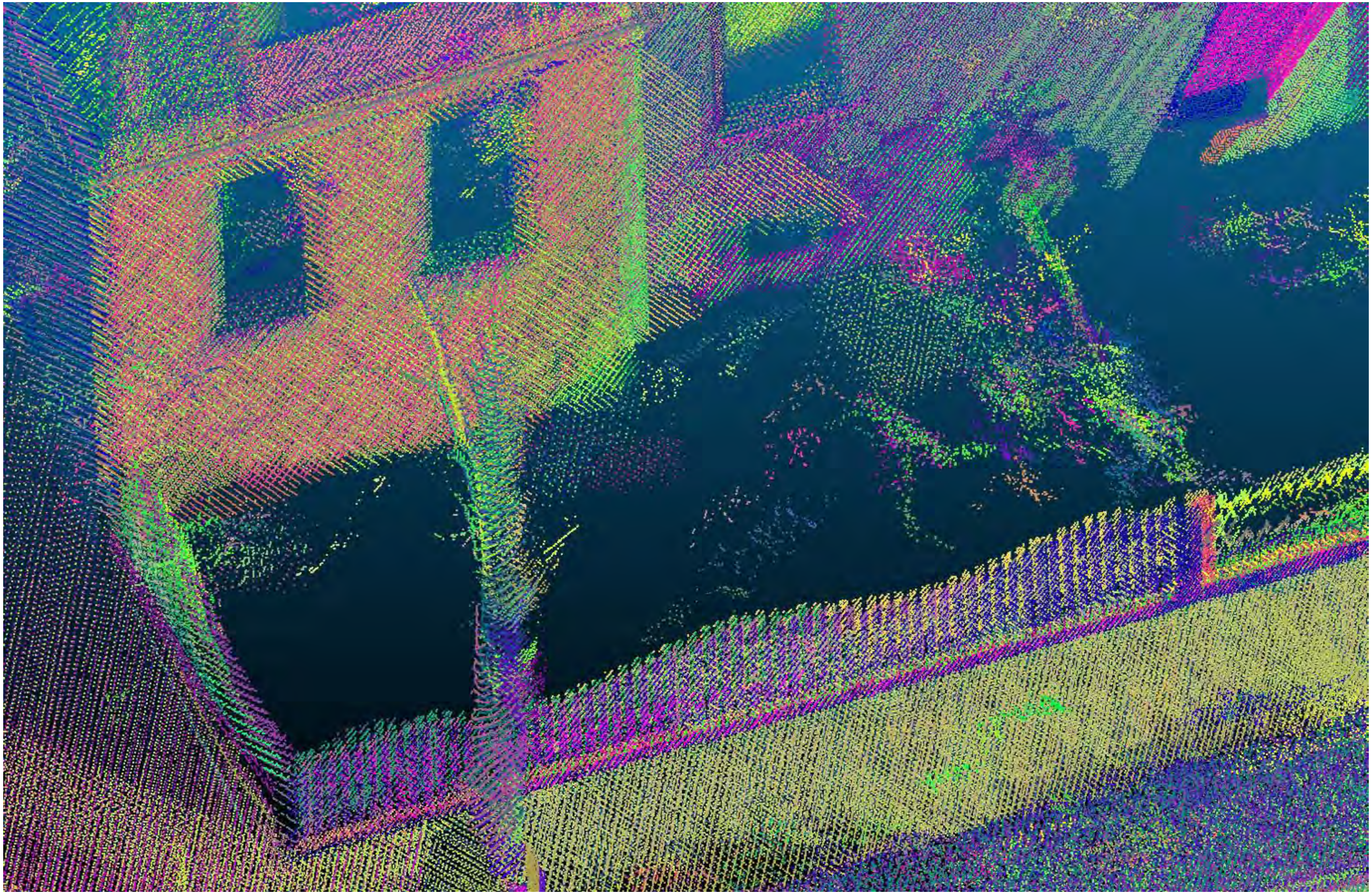


MapReduce processing scheme

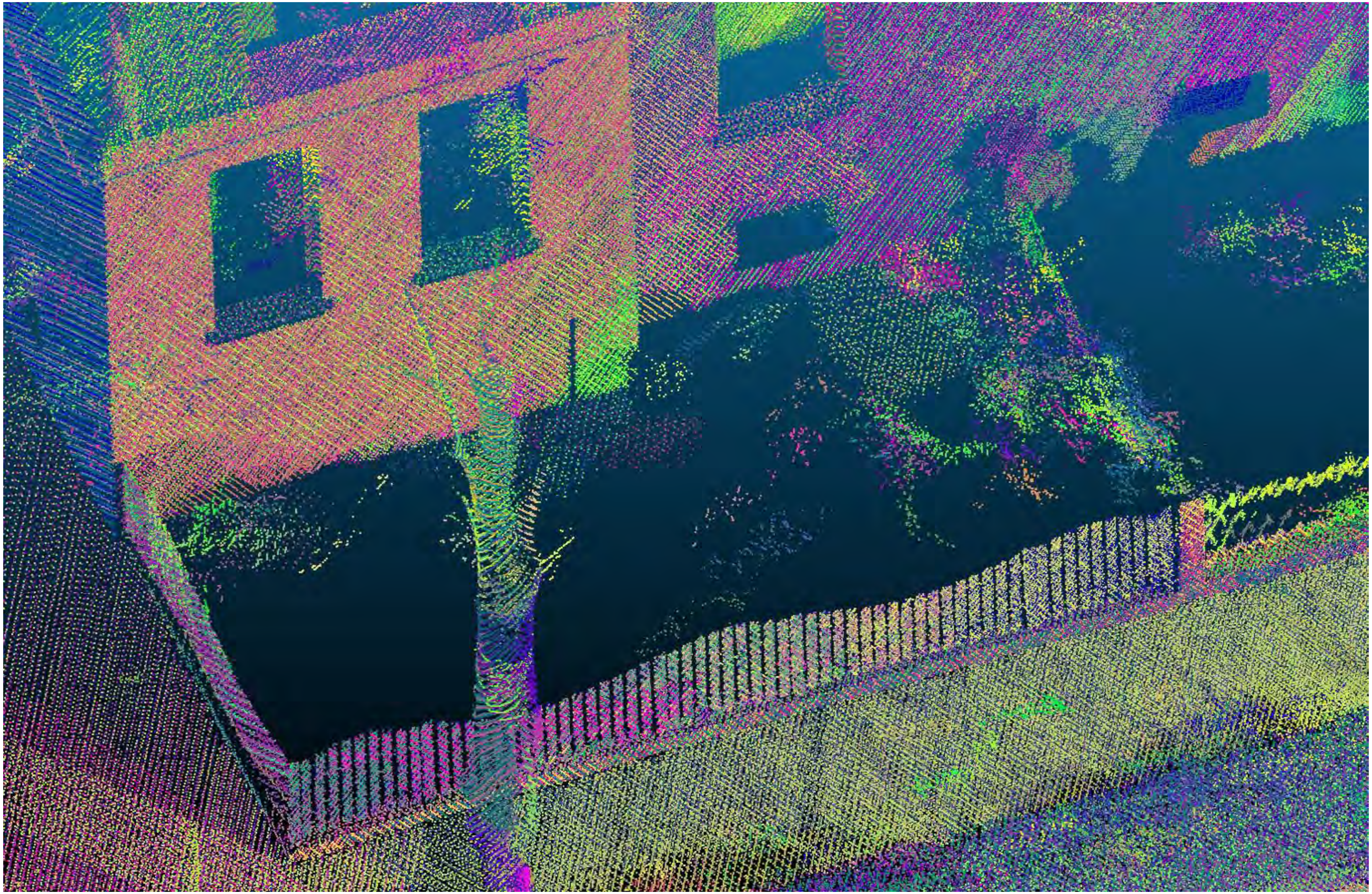
- (key, value) pairs generated by the mappers: (trajectory-id, (anchor-id, A^TPA and A^TPl sub-blocks))



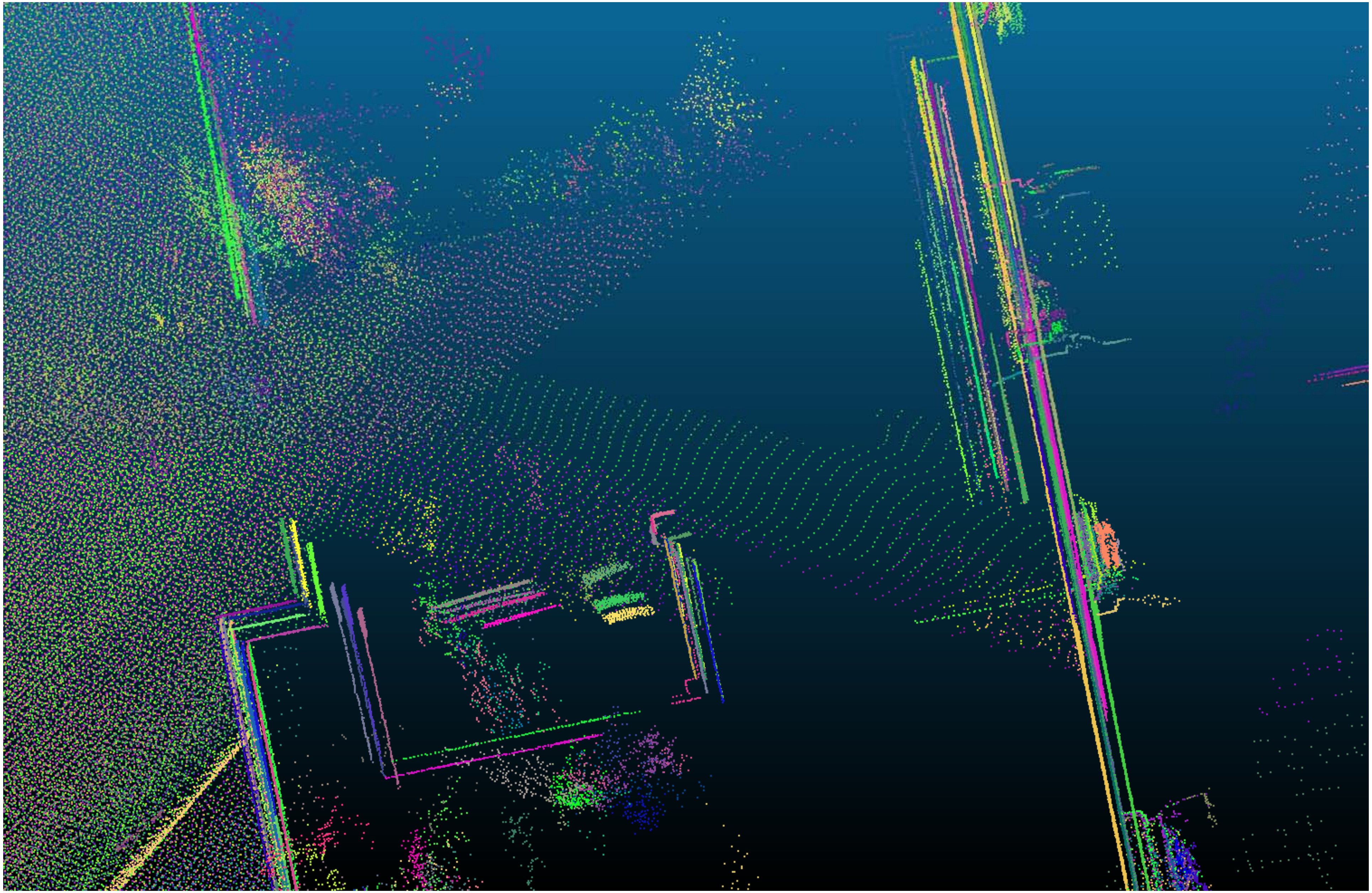
Original scans overlaid



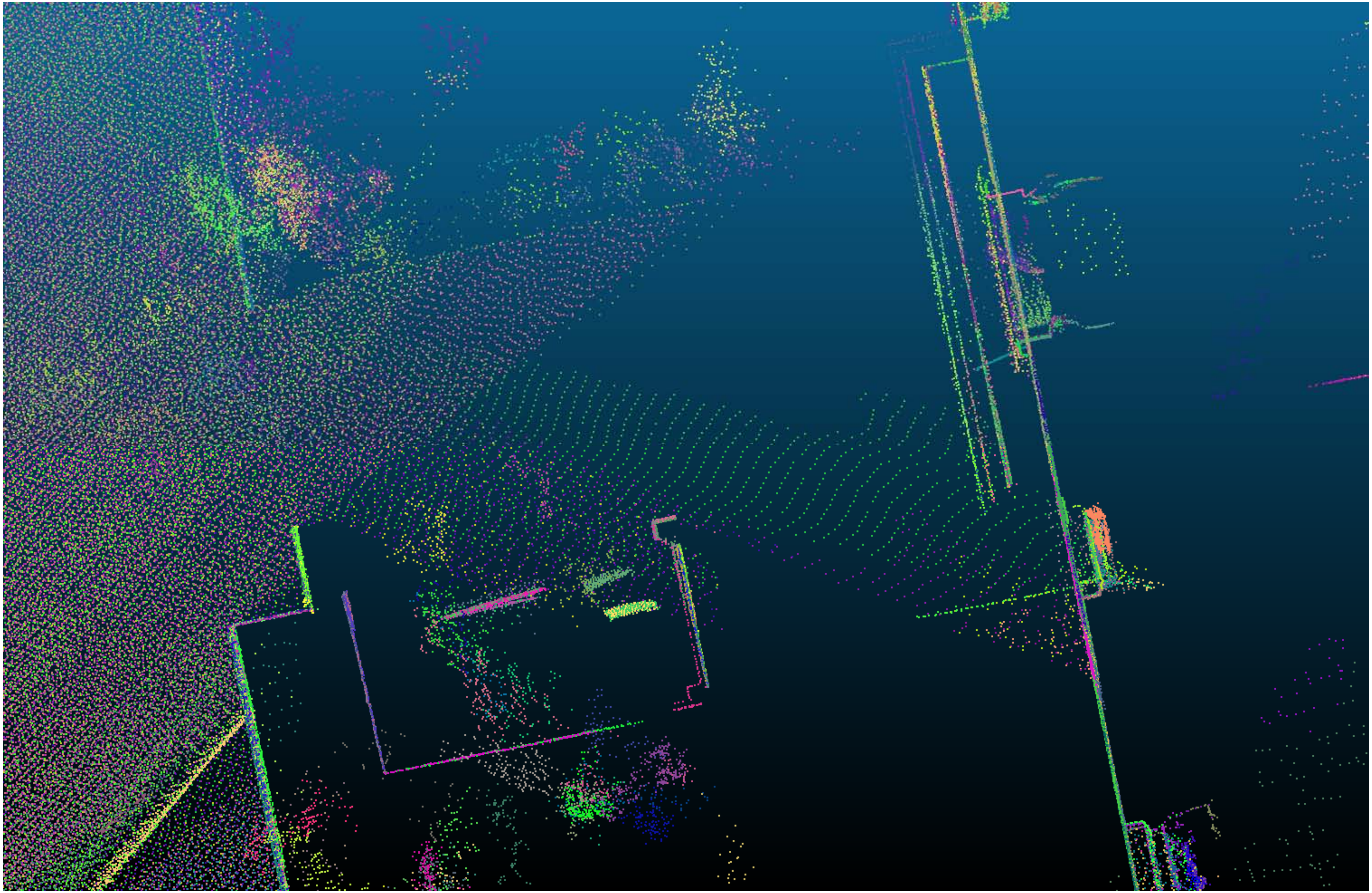
After alignment

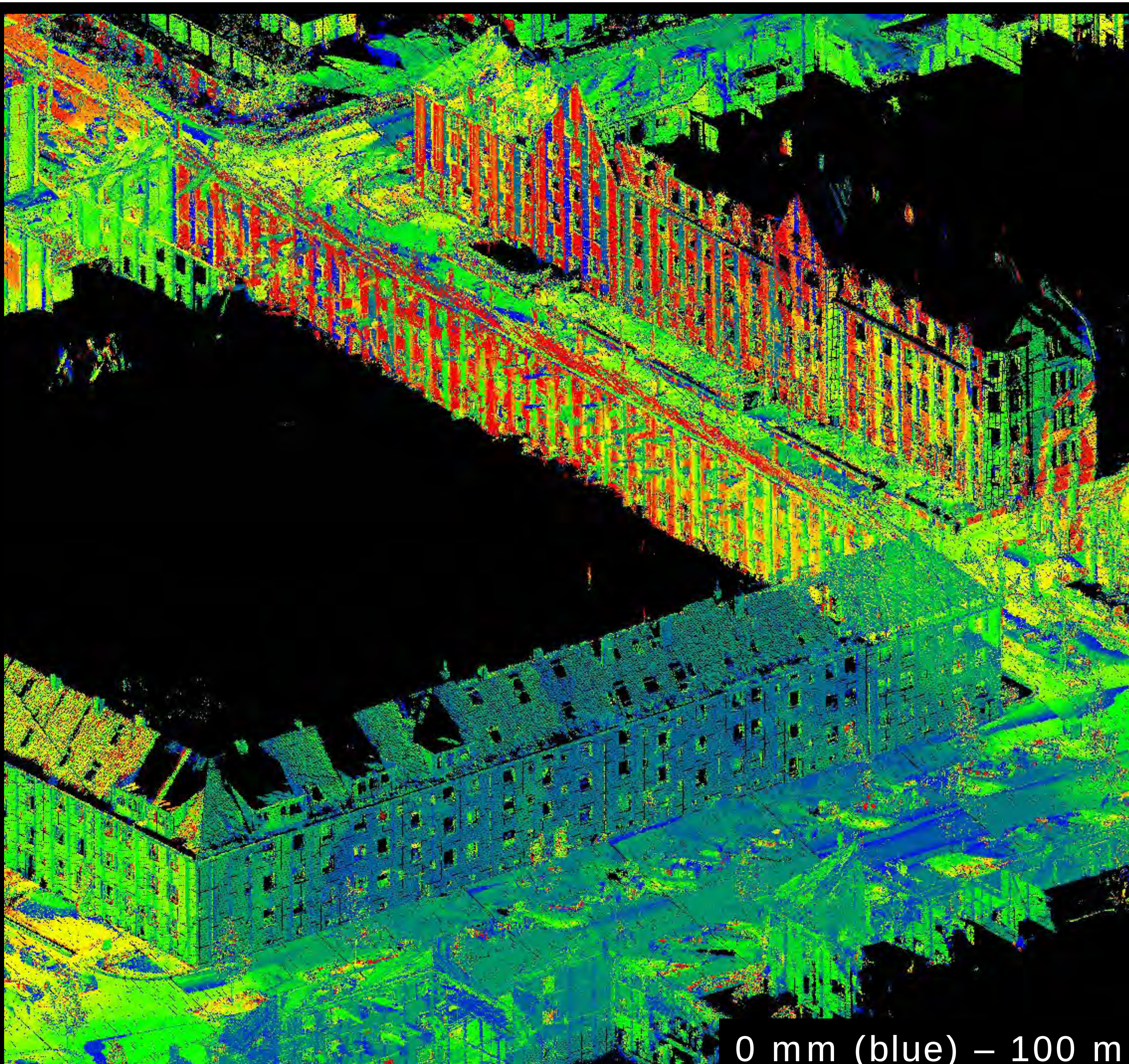


Original scans overlaid

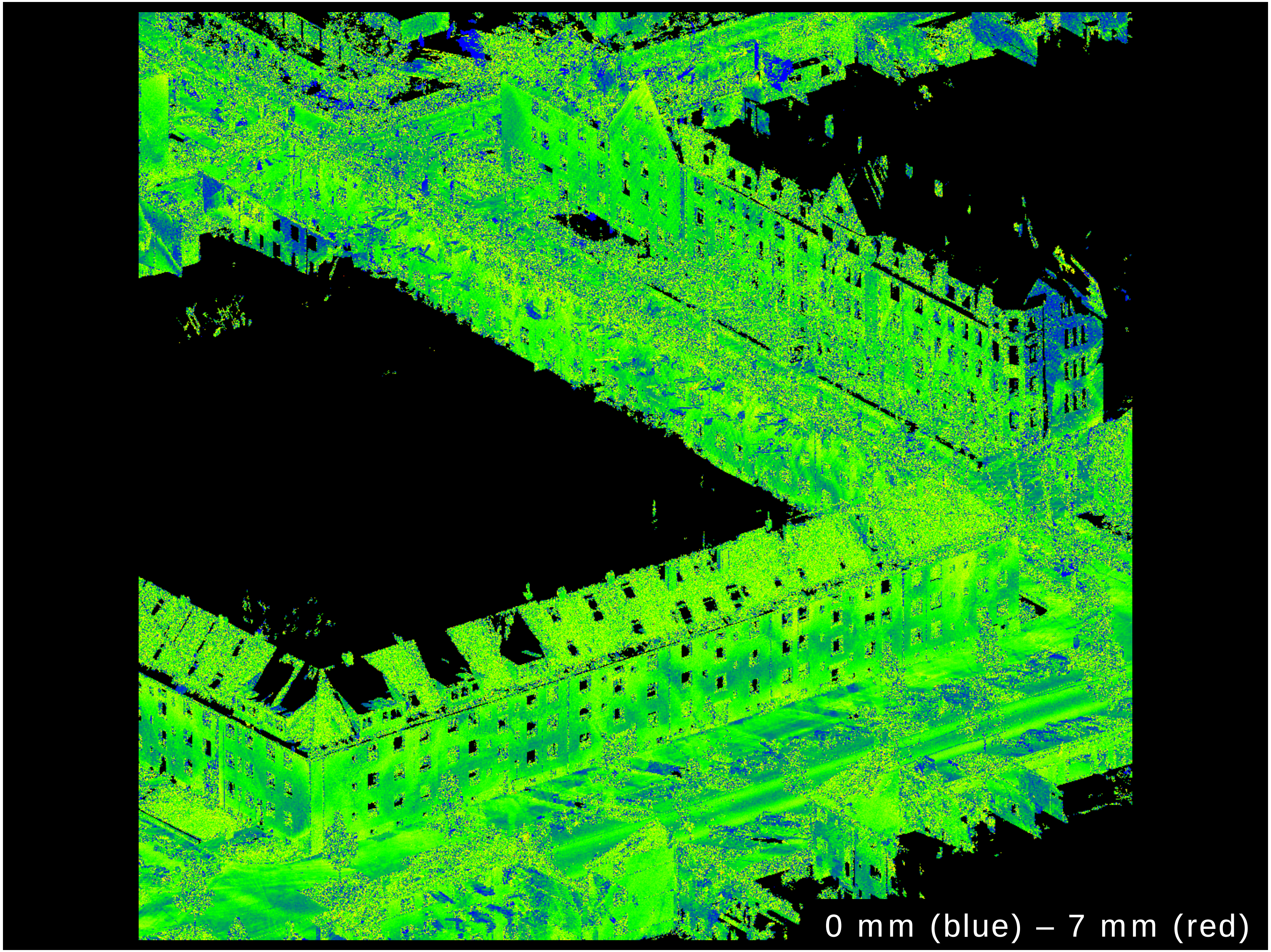


After alignment

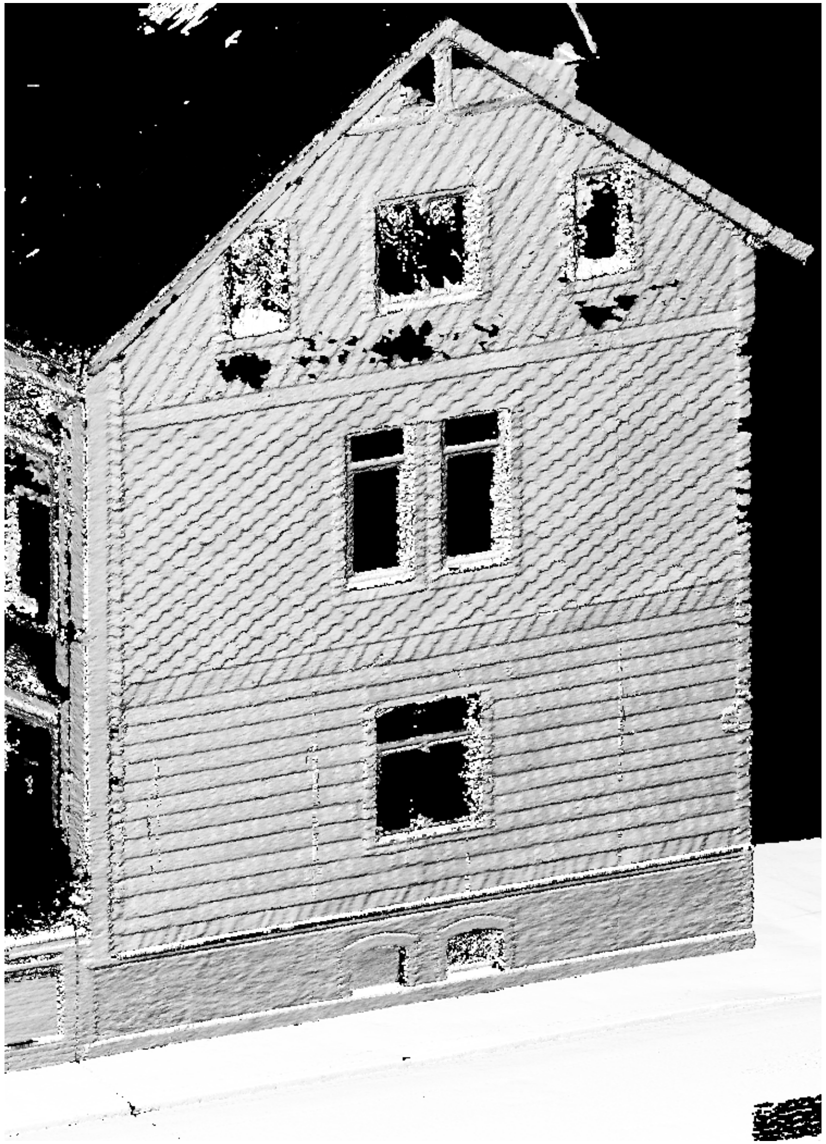




0 mm (blue) – 100 mm (red)

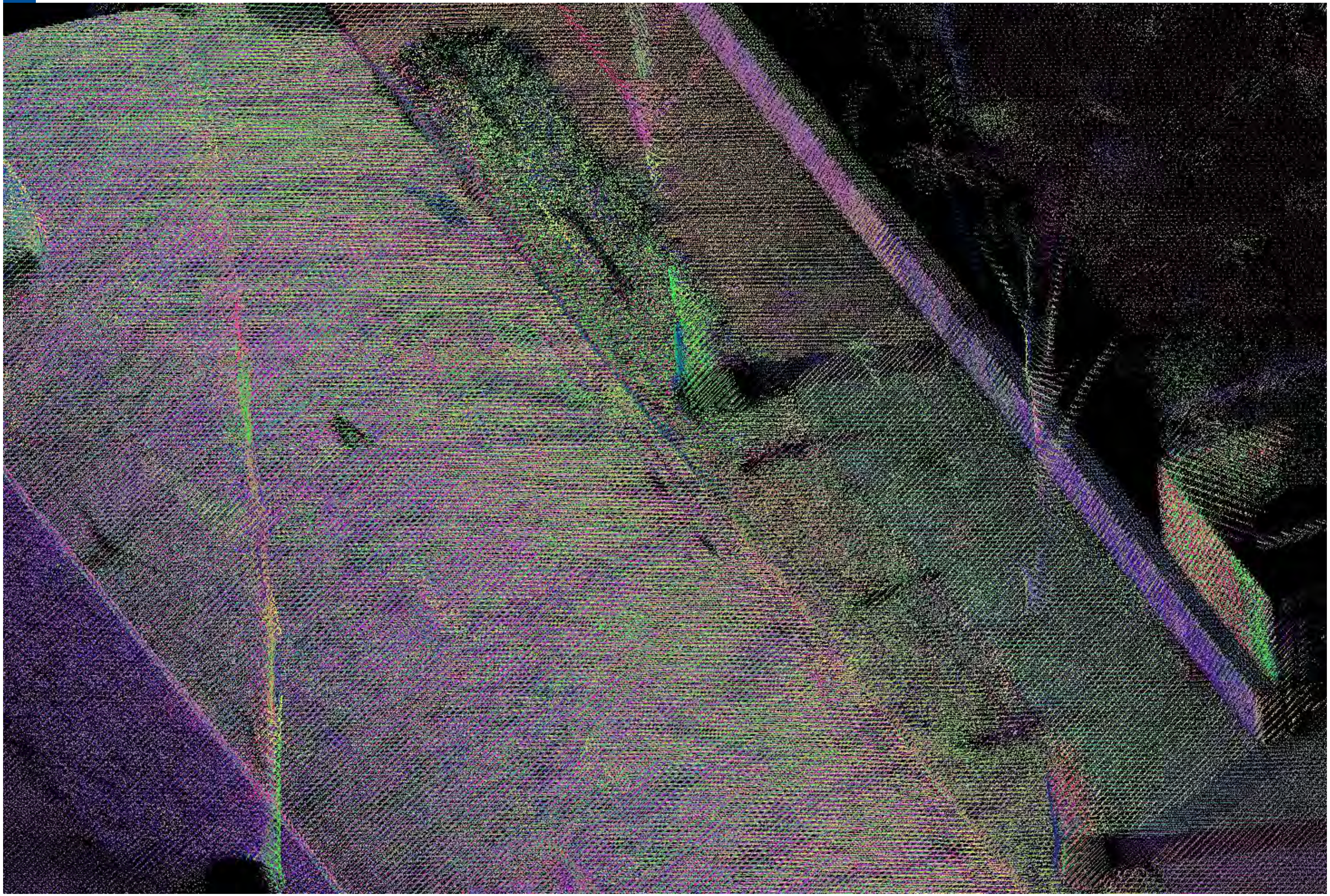

0 mm (blue) – 7 mm (red)

Latent map vs. image

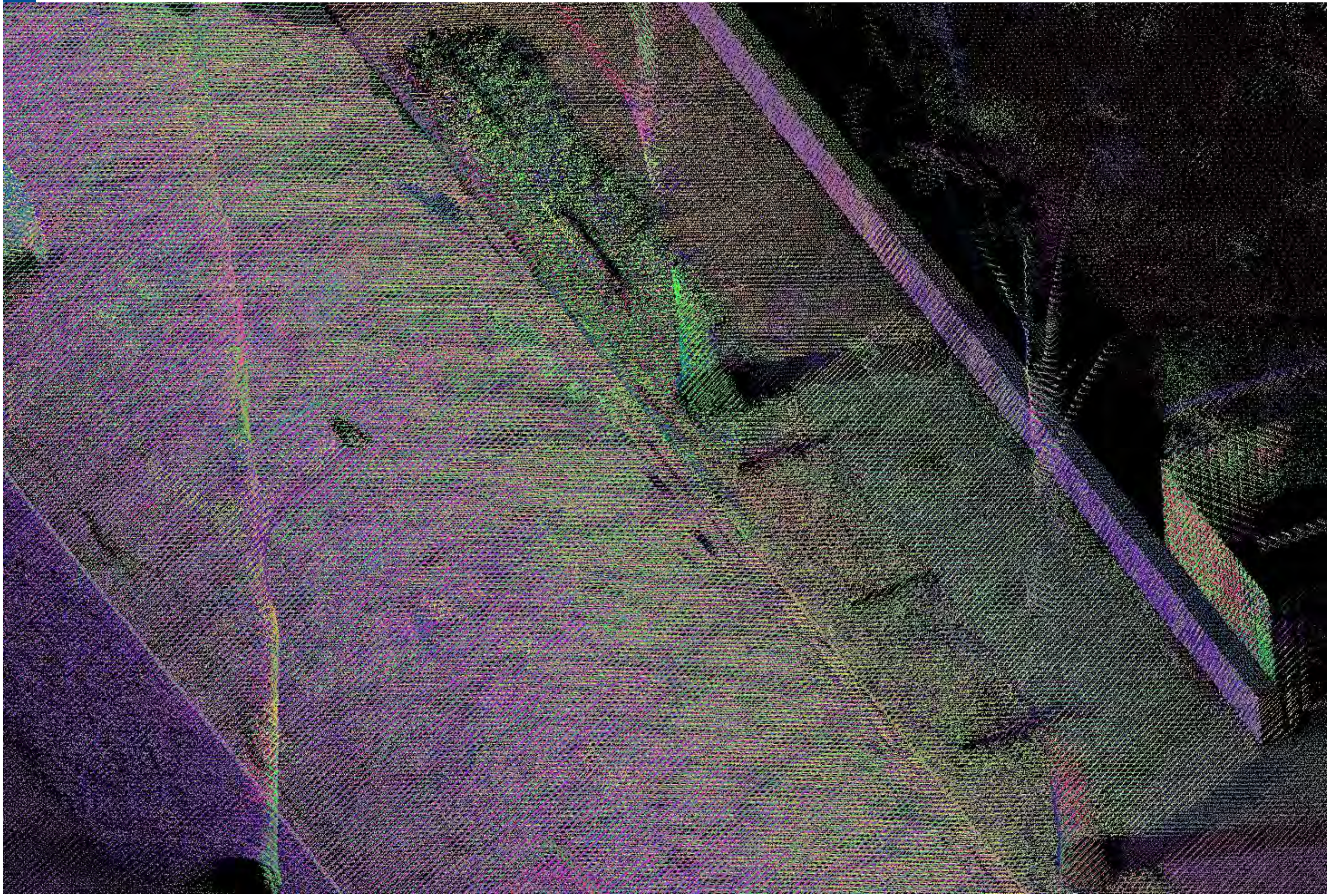


(Image source: Google StreetView)

Before adjustment



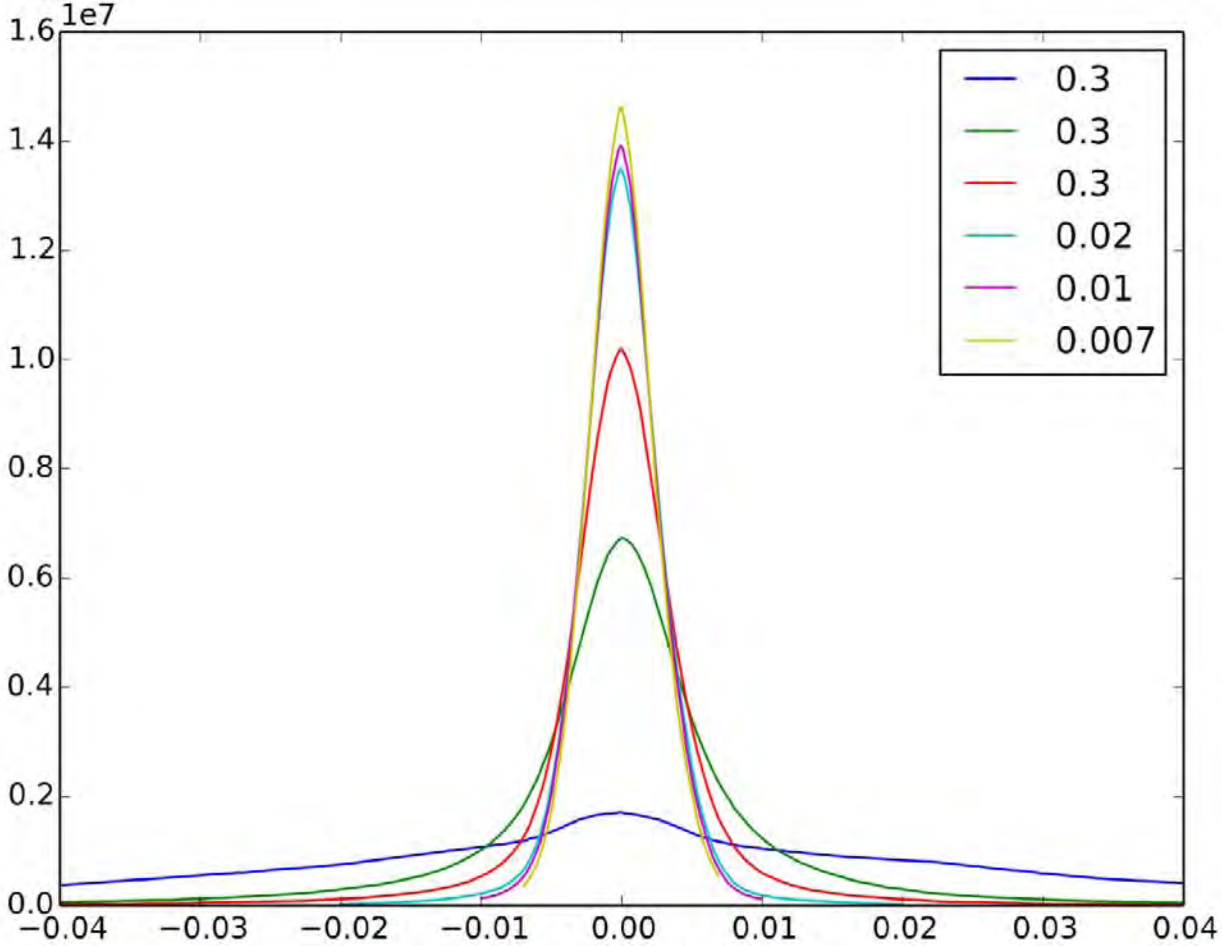
After adjustment



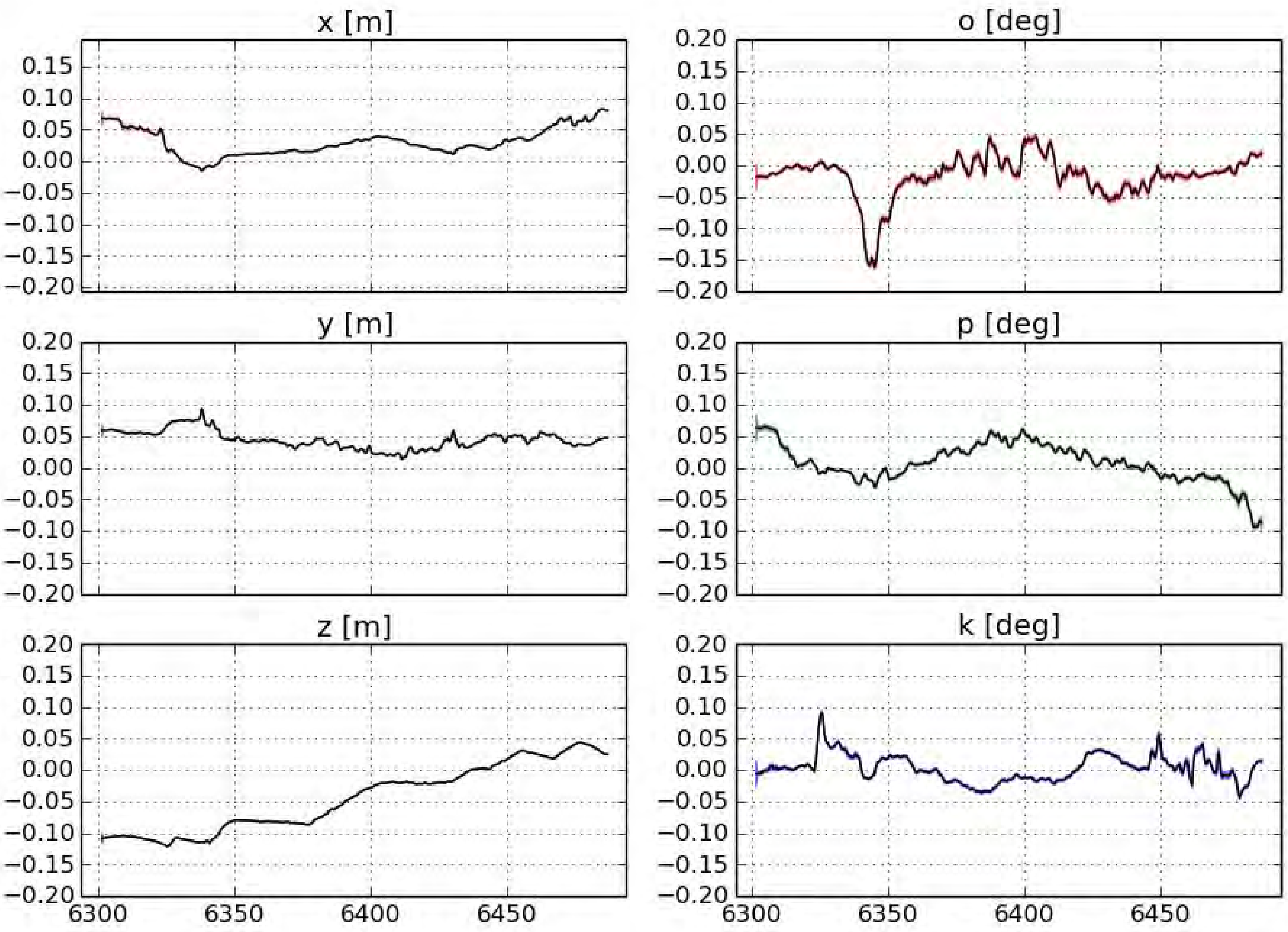
Latent map elements

Signed distance histograms

- Signed distance between points and latent map
- It's a histogram with 0.1 mm buckets!



Example: anchor estimation for one trajectory



Ricklingen dataset

(Aerial image credit: Microsoft)

Ricklingen dataset

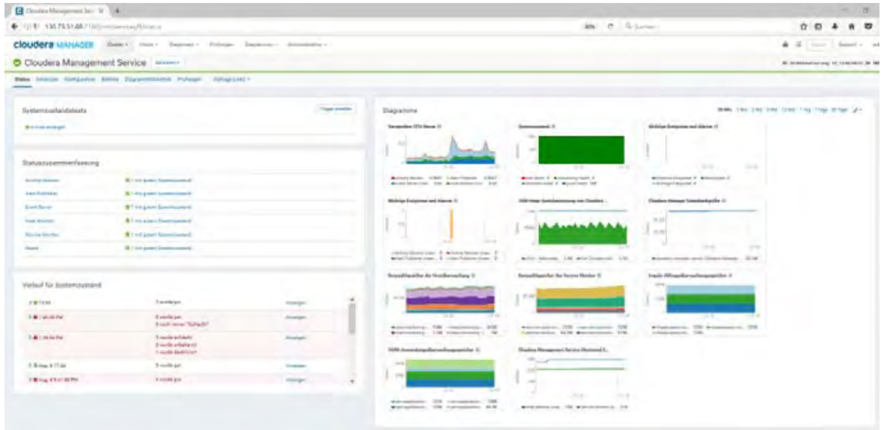
Performance

- Intel i7-4790k (4 cores, 8 threads)
- Data read from SSD
- Tiles 15 x 15 m^2

	Badenstedt	Ricklingen
Points (segmented)	1.1 bn	3.6 bn
Point data (segmented)	59 GB	199 GB
Scan strips	150	485
Number of tiles	1,287	9,588
Average / max points per tile	813 k / 18 M	371 k / 7 M
Latent map elements @ 2cm	330 M	1.9 bn
Estimated orientation unknowns	278,052	780,780
Processing @ 2 cm (one iteration)	710 s	2,260 s
Overall processing: points/s	1.5 M	1.6 M

Hadoop Cluster since 2017 (RTG i.c.sens)

- 6 x 2 CPUs, 96 Cores
- 768 GB RAM, 288 TB HDD
- 10 Gbit switch (2x10/server)
- Extra RAID file server
- Cloudera distribution

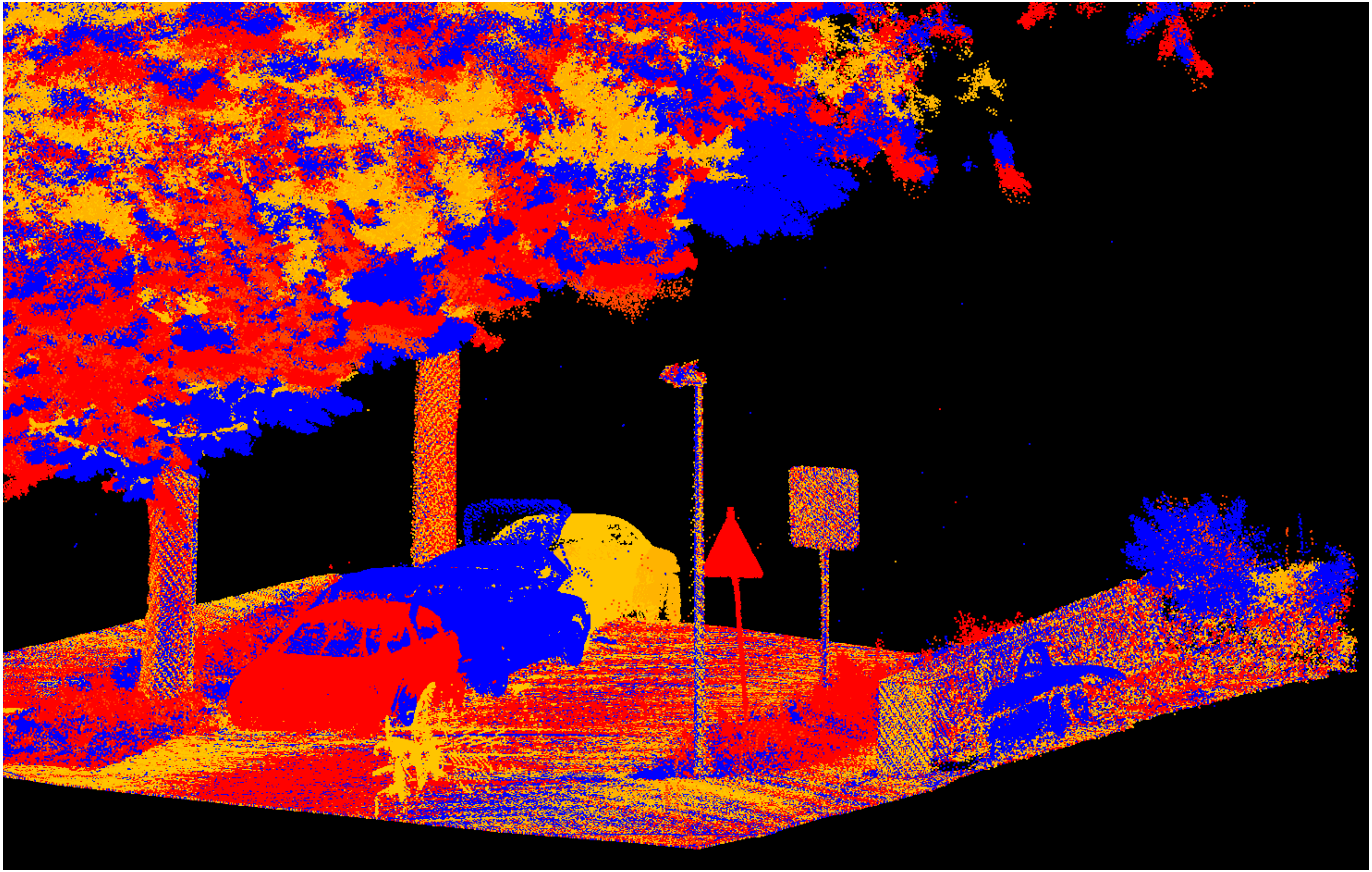


Dynamic maps: change detection

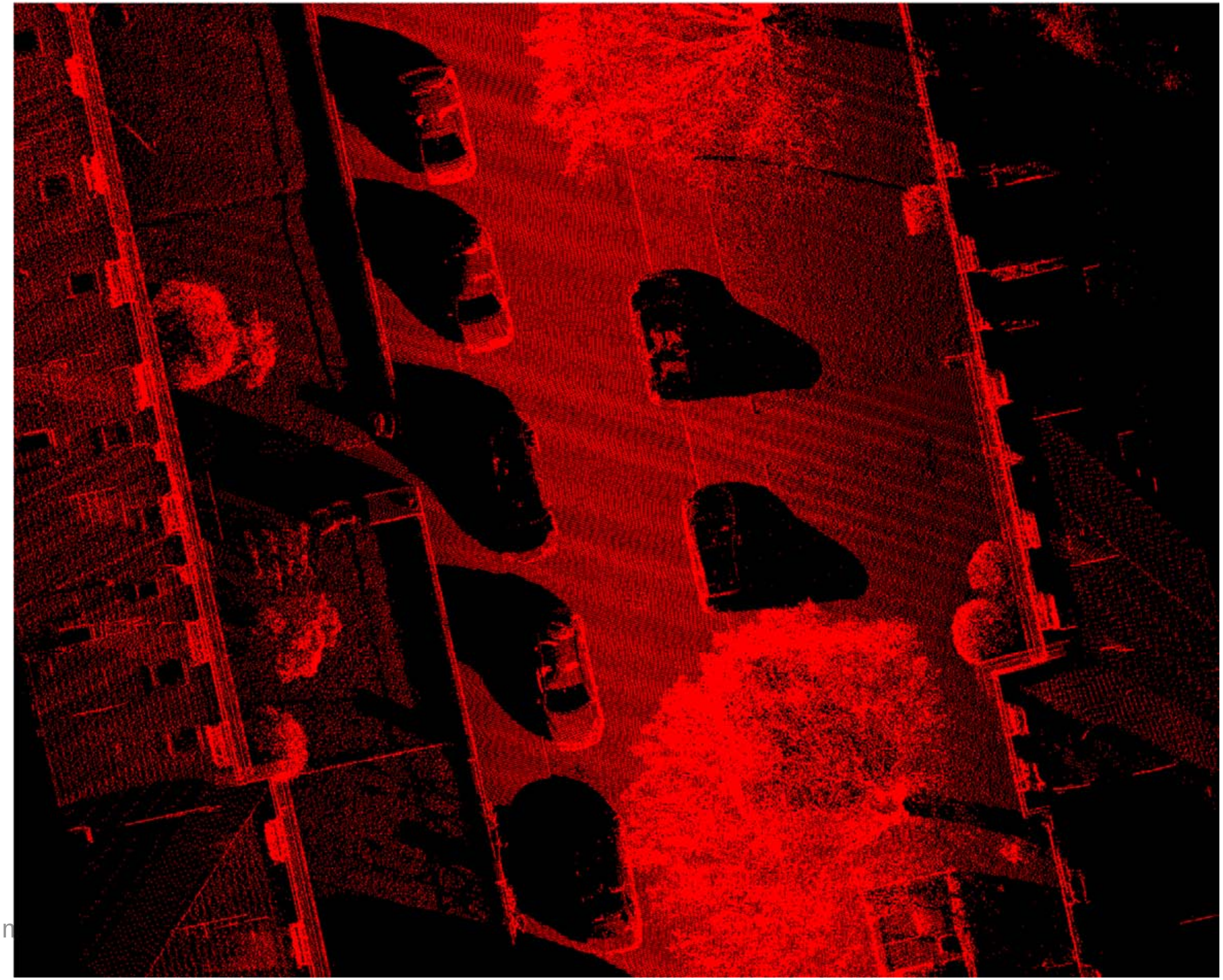
Change detection and dynamic maps

- A hard problem: **long-term mapping**
- The world is not static, how to deal with changes?
  - **Stability-plasticity dilemma**
  - Naive approaches: static map, variable obstacles
- Idea:
  - Identify **latent variables that control "change"**
  - Allows prediction of dynamic "world state"
  - Less "surprise" for the robot during localization and scene understanding
- Experiments:
  - 3D occupancy grid, full LiDAR ray tracing
  - Finding temporal patterns

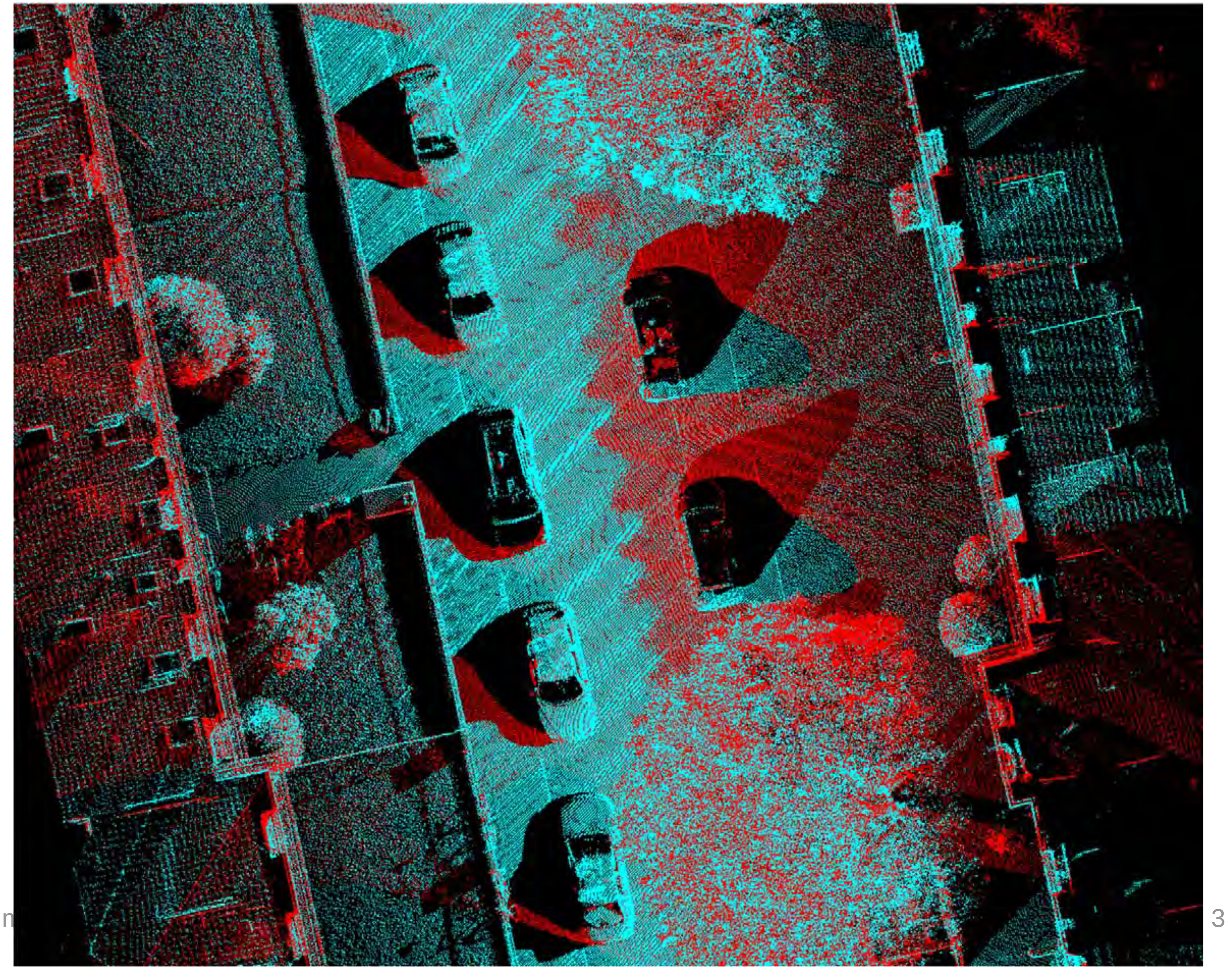
Not so small changes...



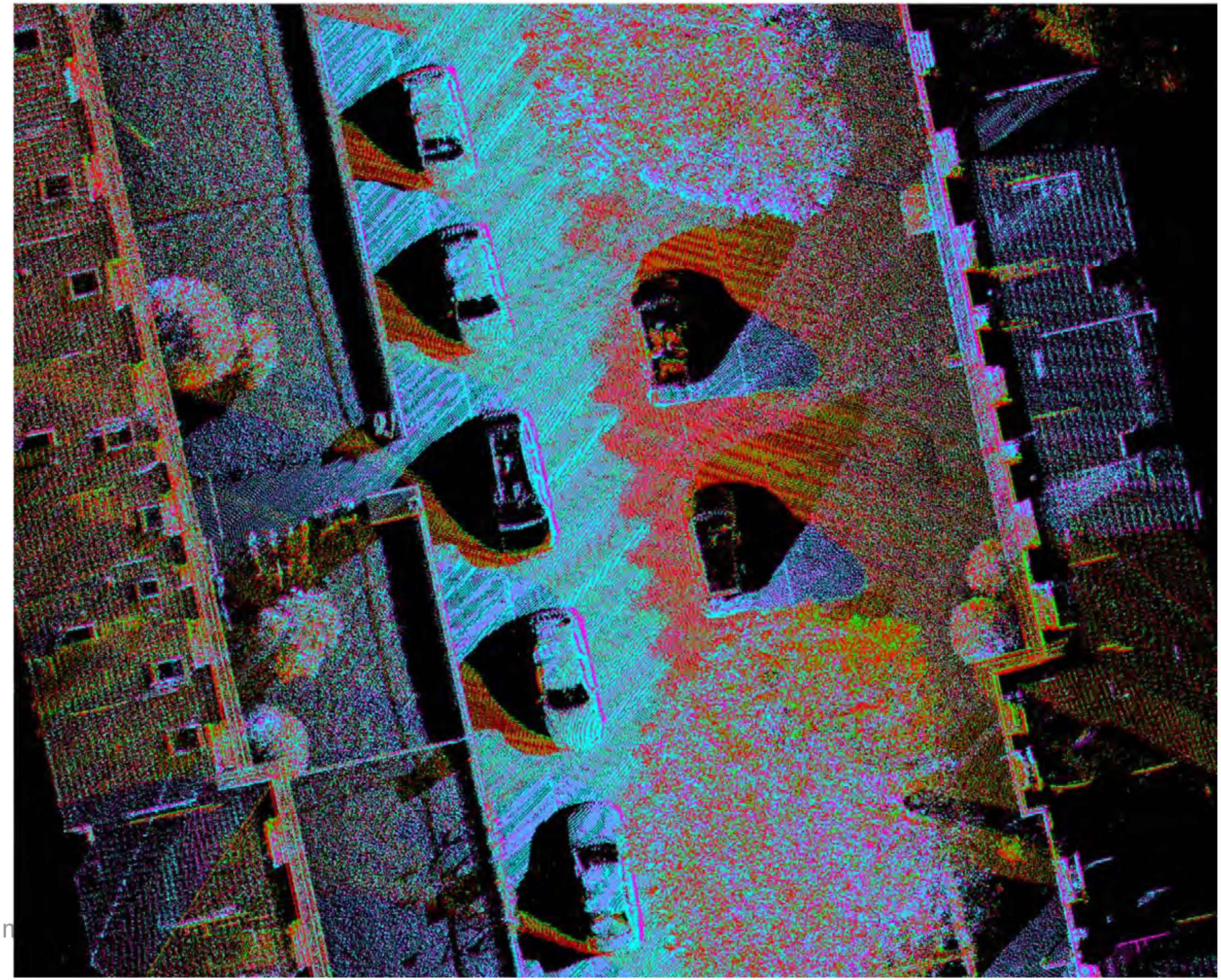
Geometry example: LiDAR scans: 1 (I)



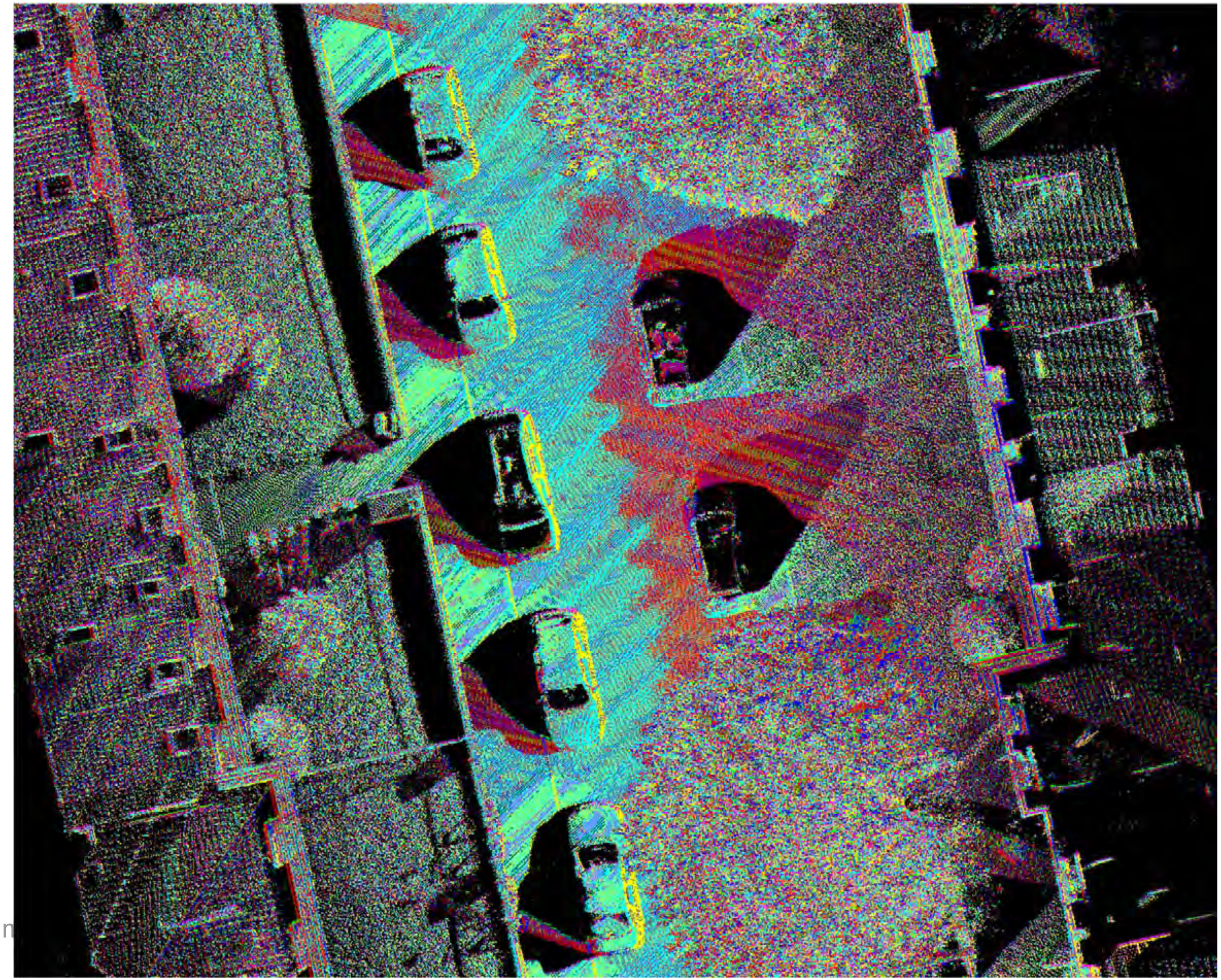
Geometry example: LiDAR scans: 1 (l+r)



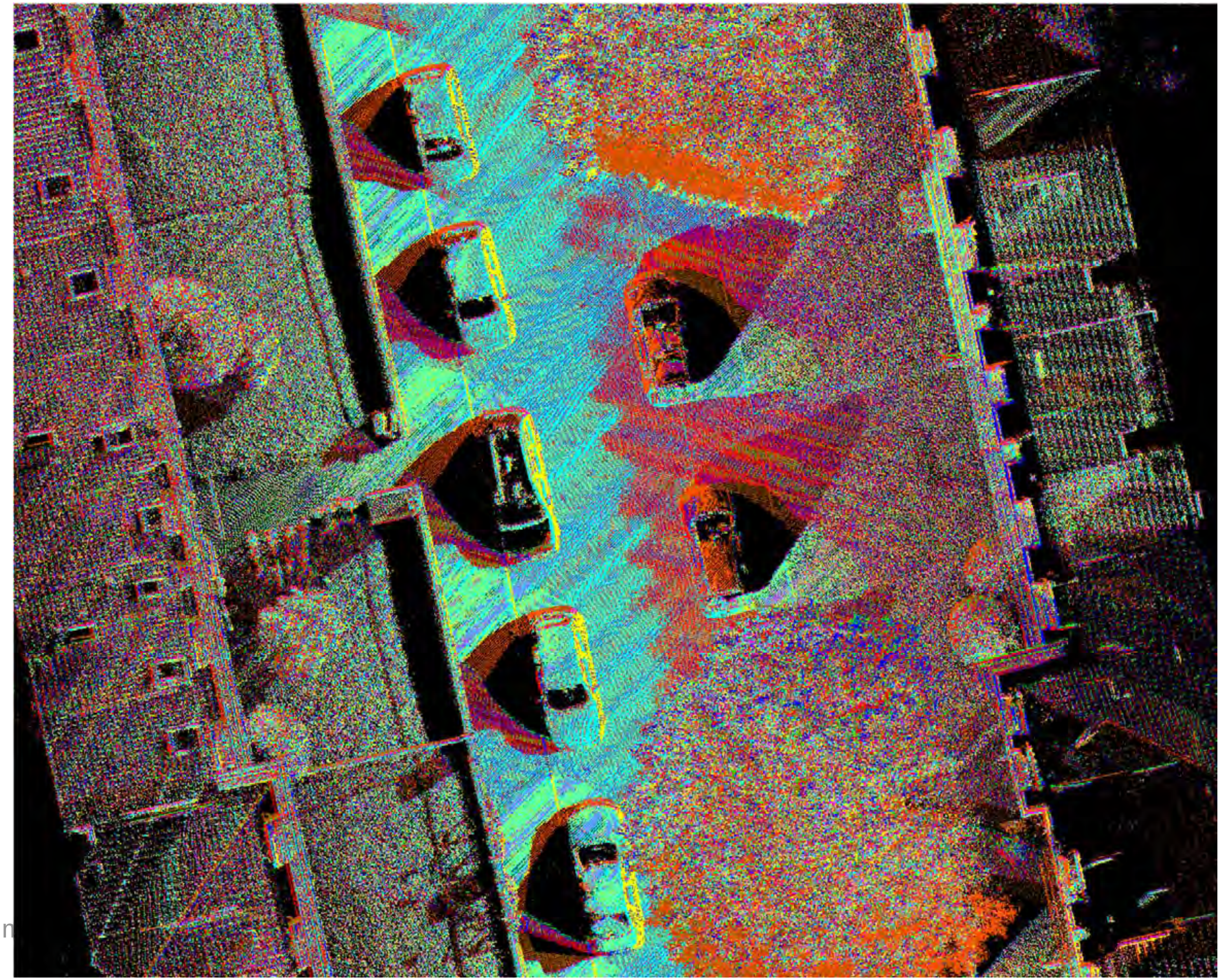
Geometry example: LiDAR scans: 12



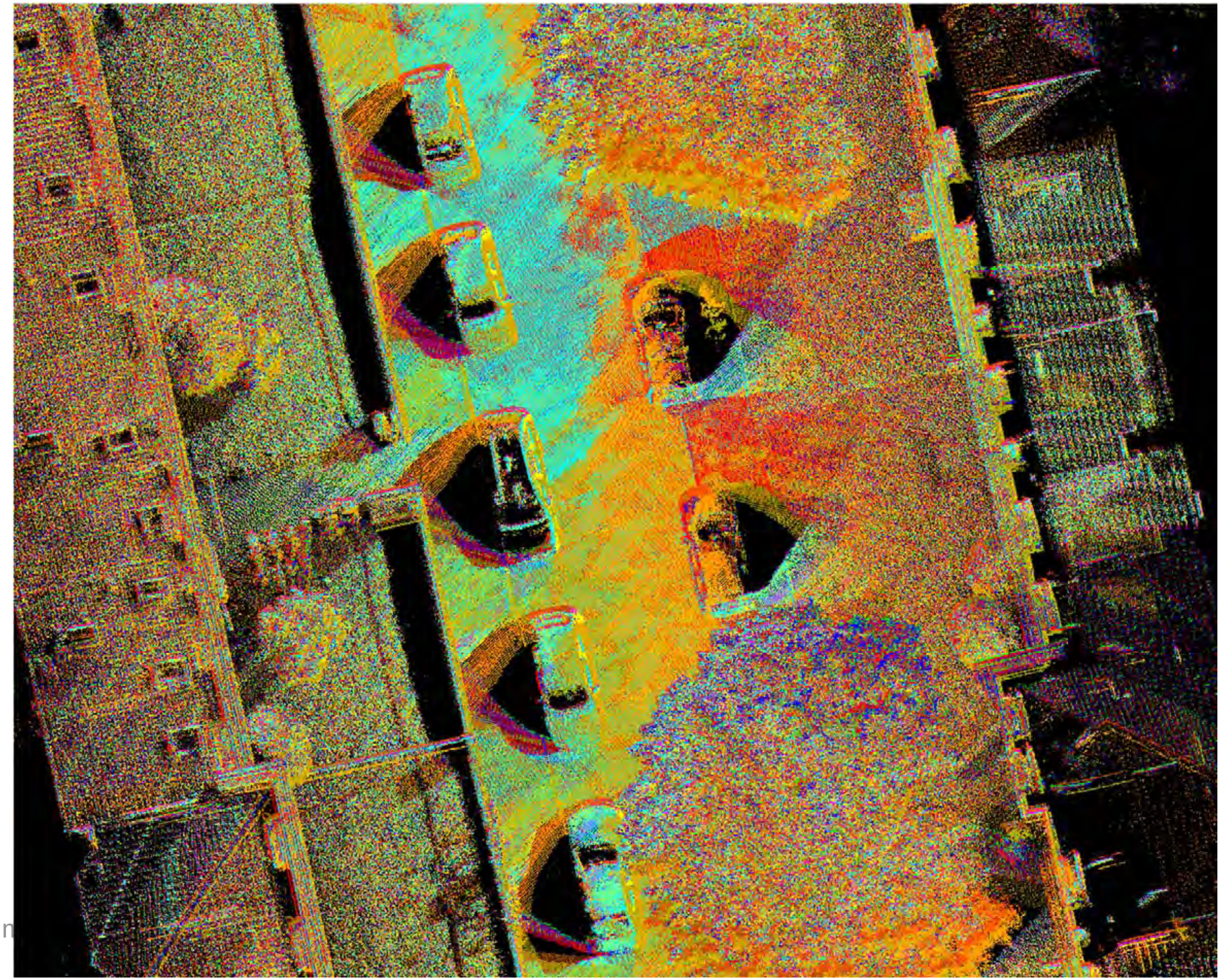
Geometry example: LiDAR scans: 123



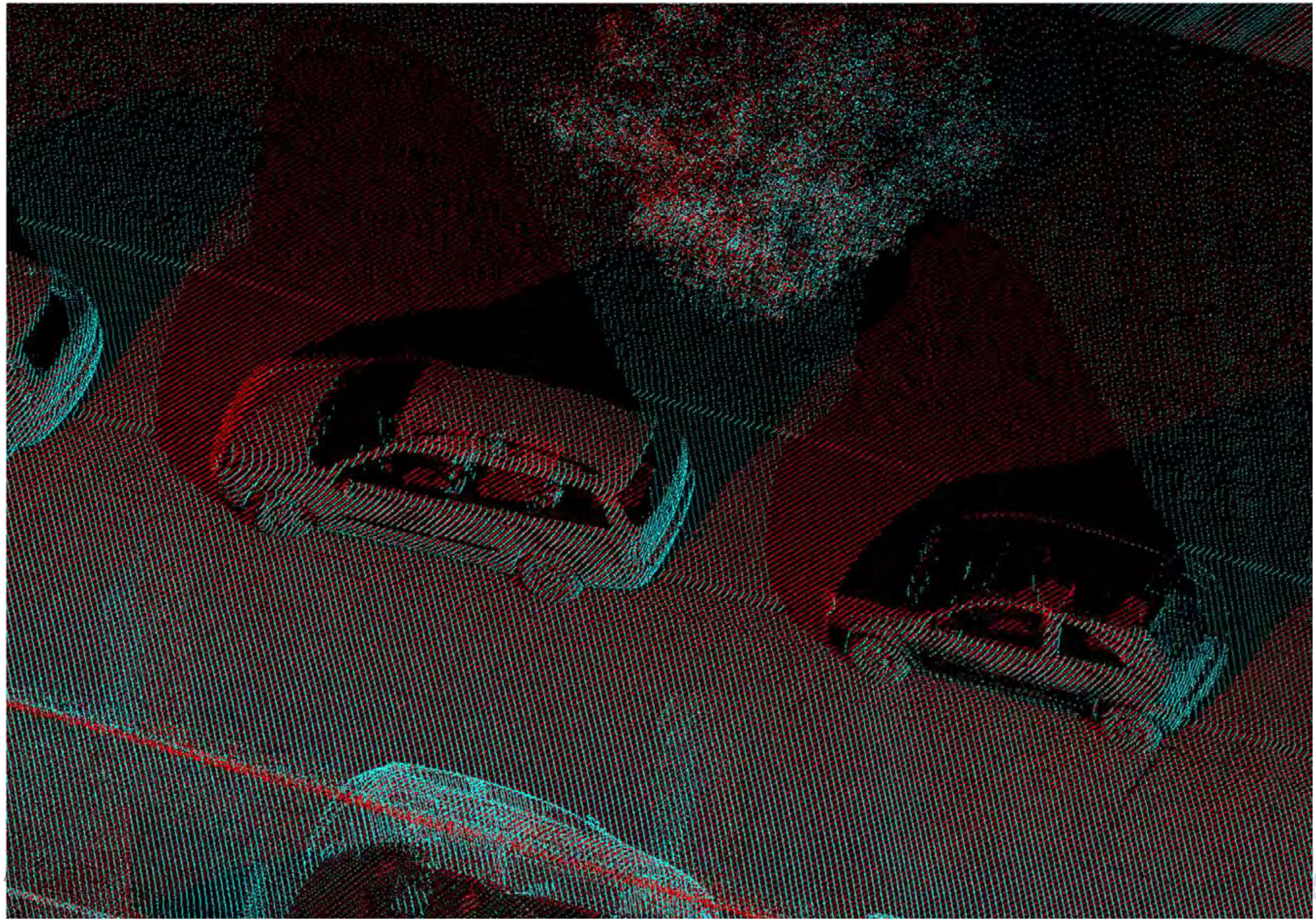
Geometry example: LiDAR scans: 1234



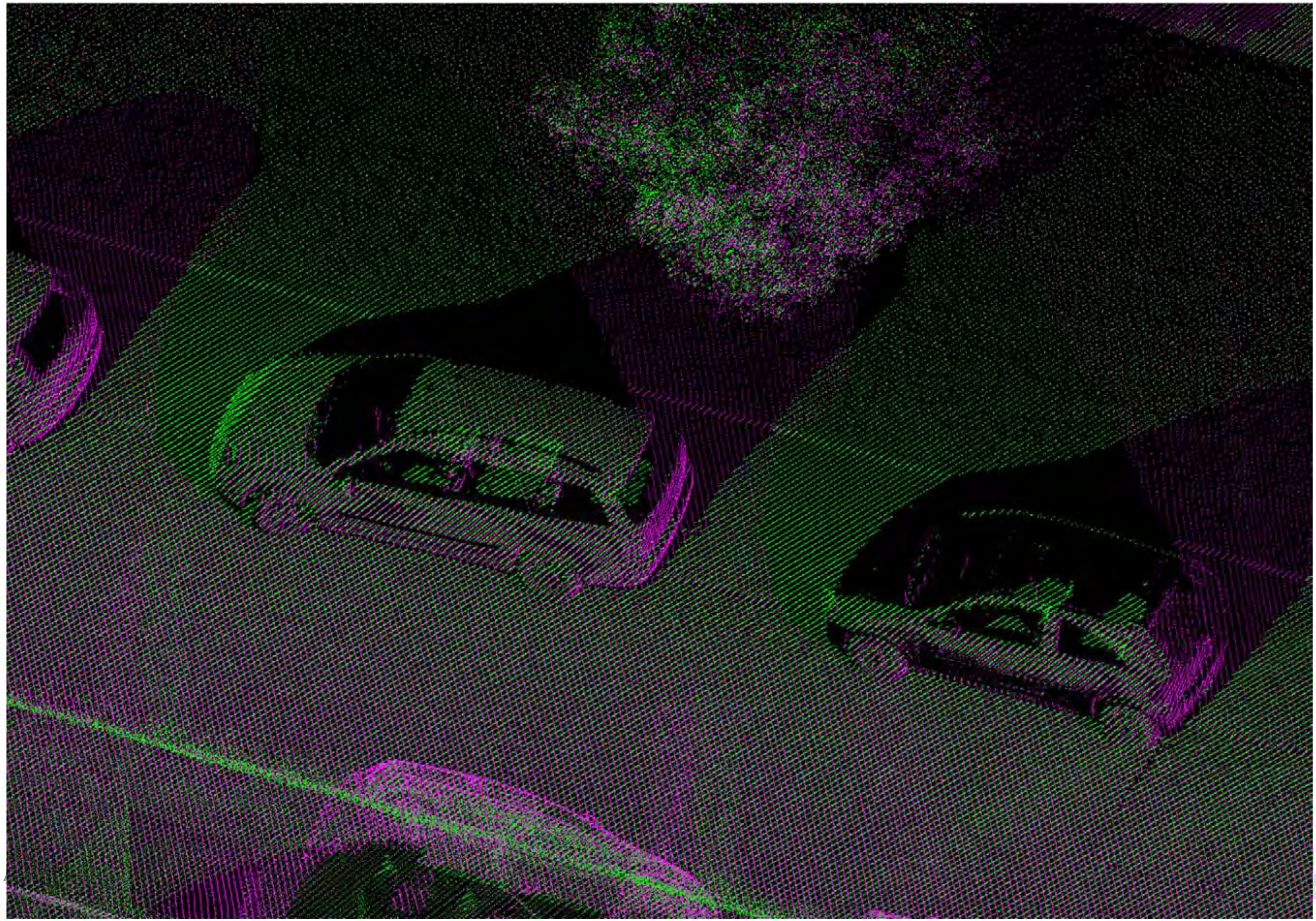
Geometry example: LiDAR scans: 12345



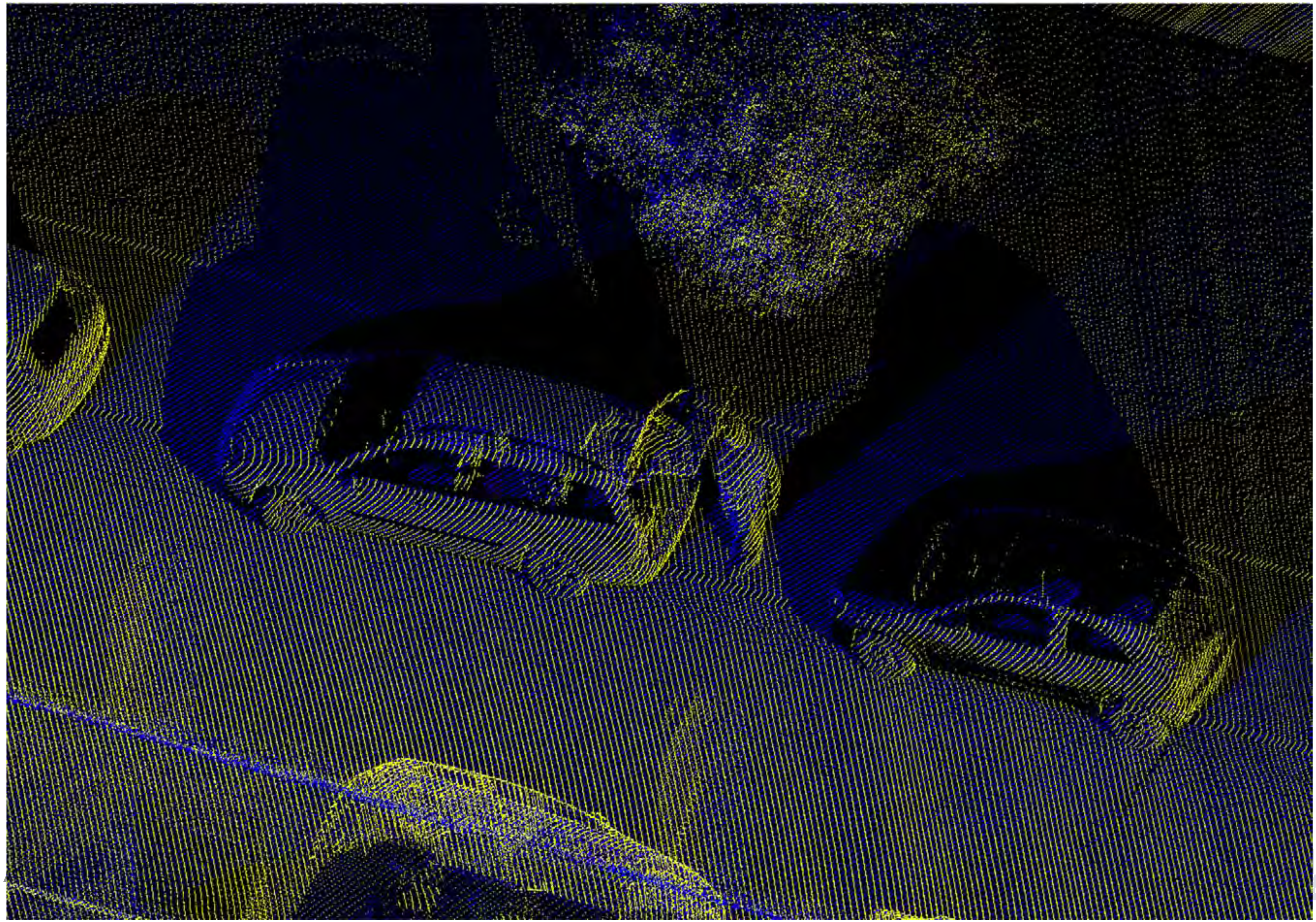
Geometry example: LiDAR scans: 1 (l+r)



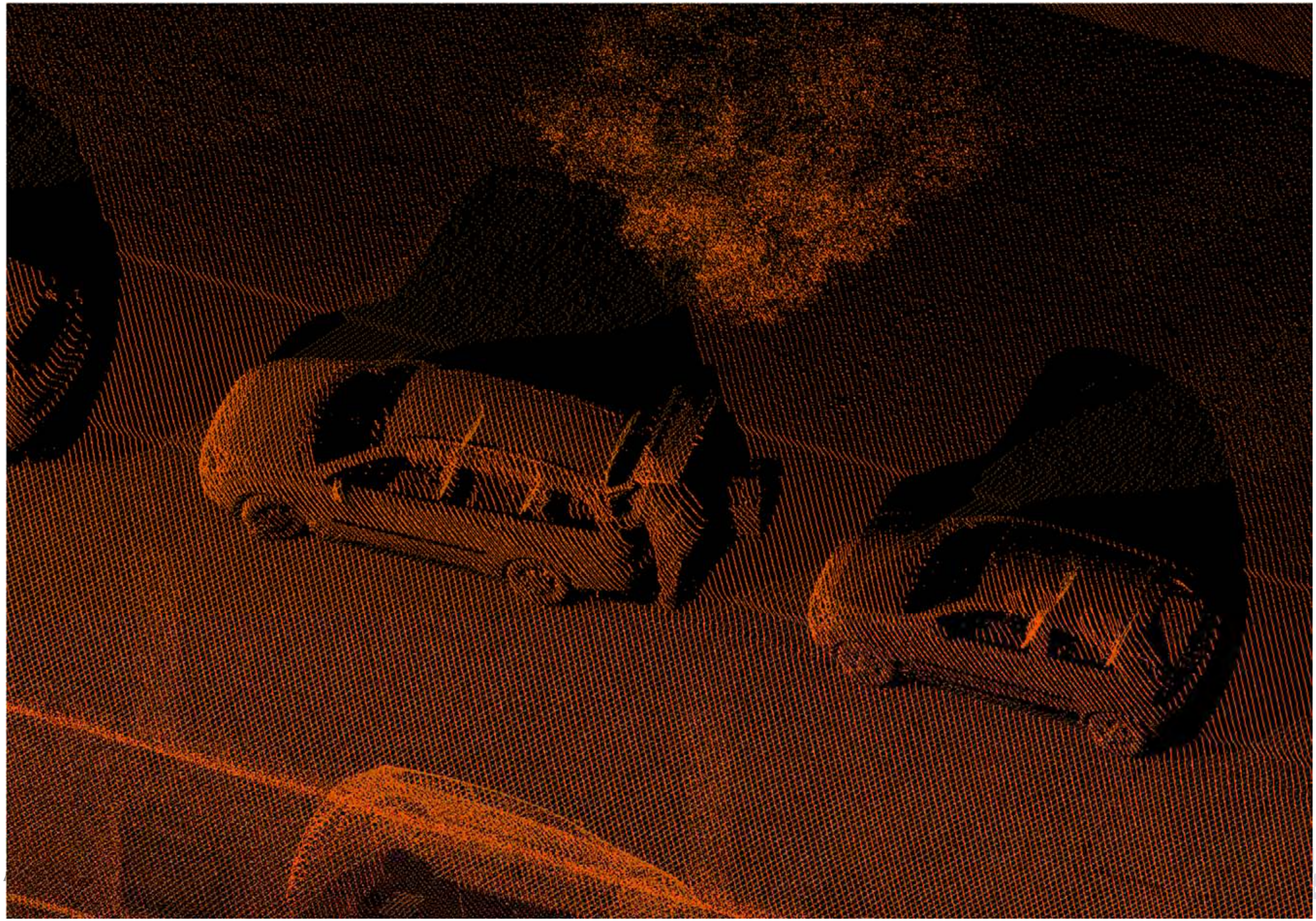
Geometry example: LiDAR scans: 2



Geometry example: LiDAR scans: 3



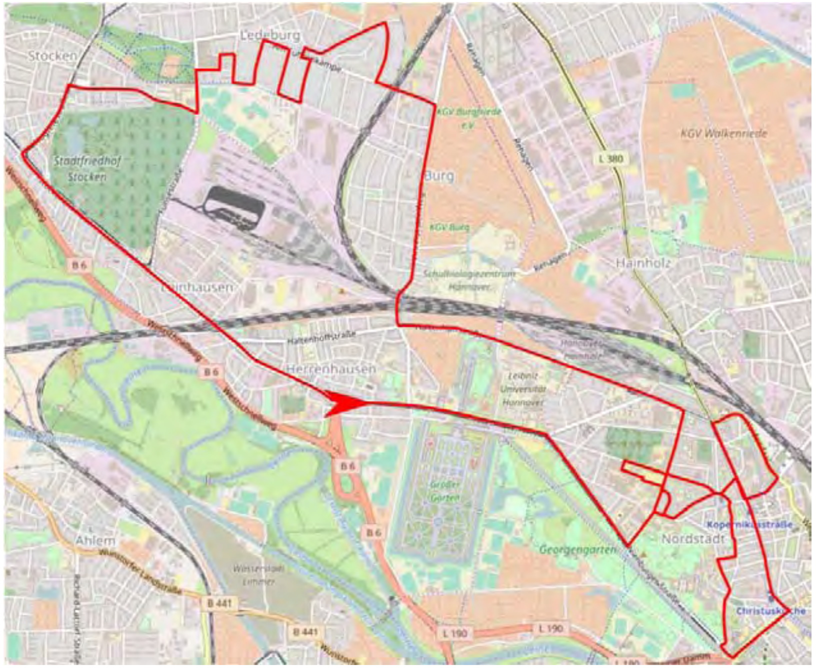
Geometry example: LiDAR scans: 4



Geometry example: LiDAR scans: 5

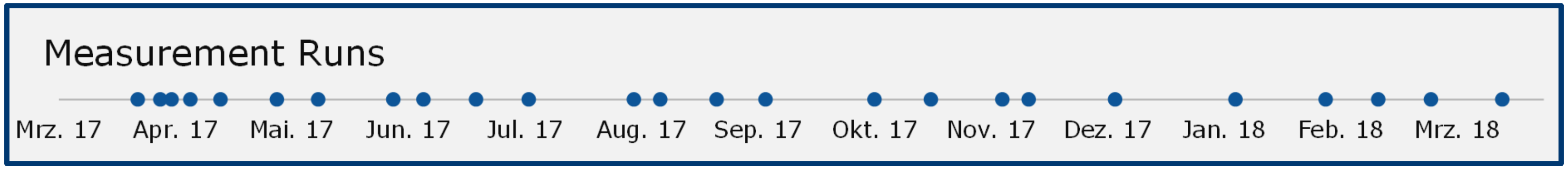
Long Term Data Acquisition

- ~20 km route in Hannover City
  - Nordstadt
  - Stöcken
  - Leinhausen
  - Herrenhausen



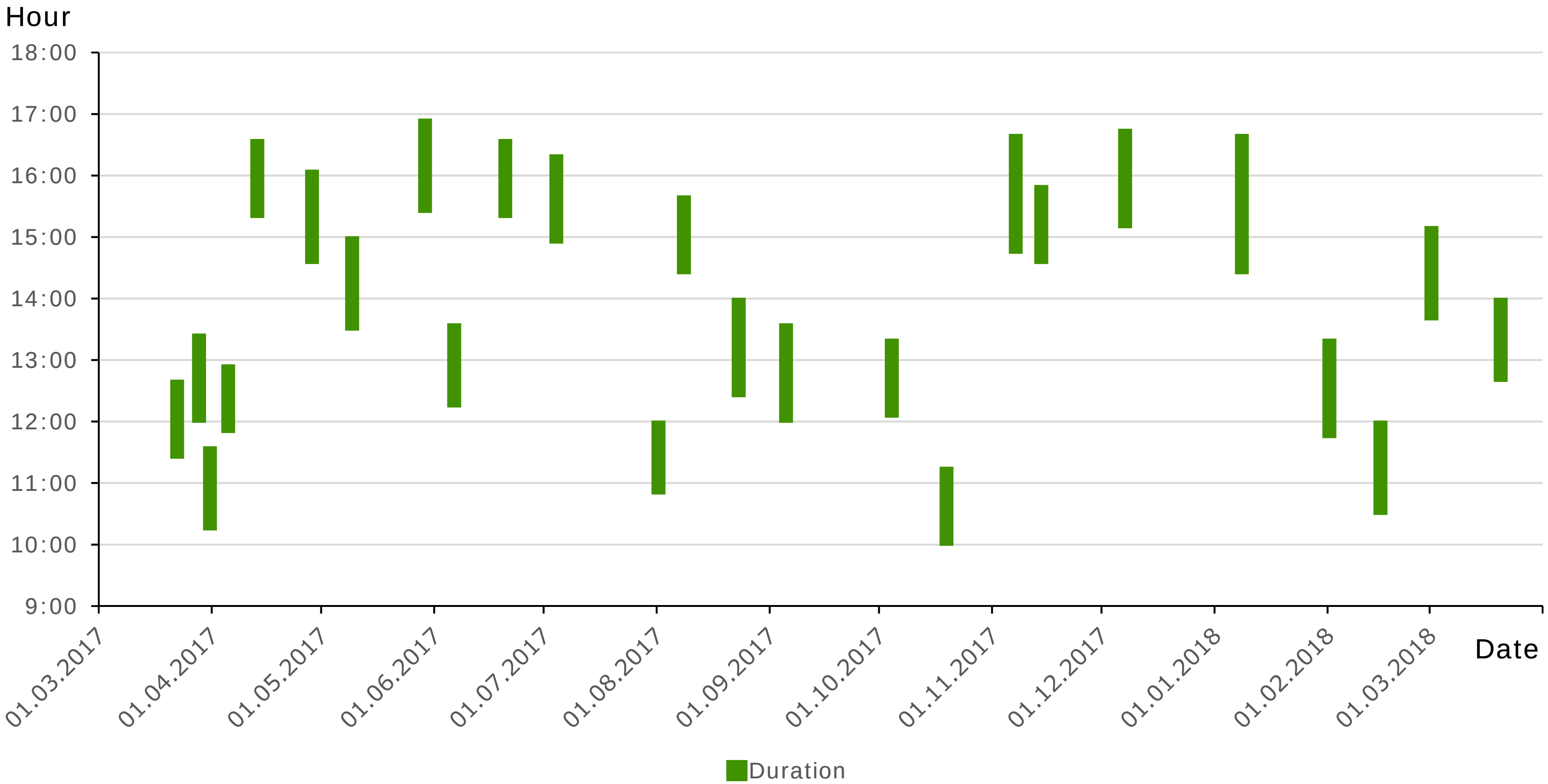
20 km Route for biweekly measurements

- One year of biweekly measurements



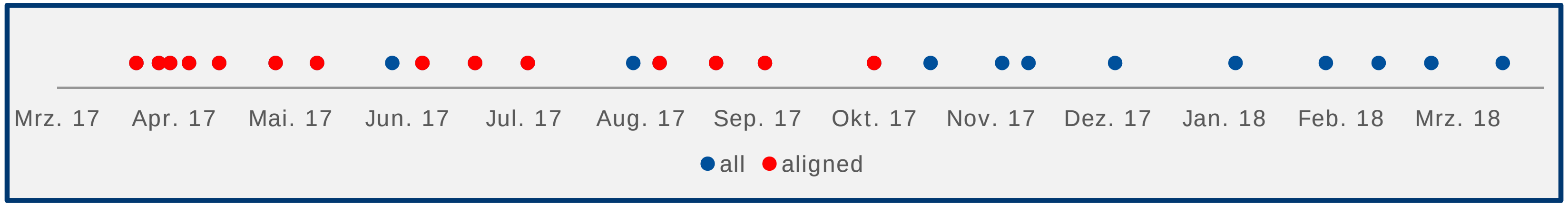
Long Term Data Acquisition

- 25 measurement runs in total
- March 2017 until March 2018,
- Different times of the day/ days of the week



Pre-Processing

- 14 measurement runs processed and aligned
- 1457 Scanstrips
- 15 017 586 980 Points
- Sort points into voxels
  - Point in a voxel is a hit
  - Ray going through a voxel is a miss



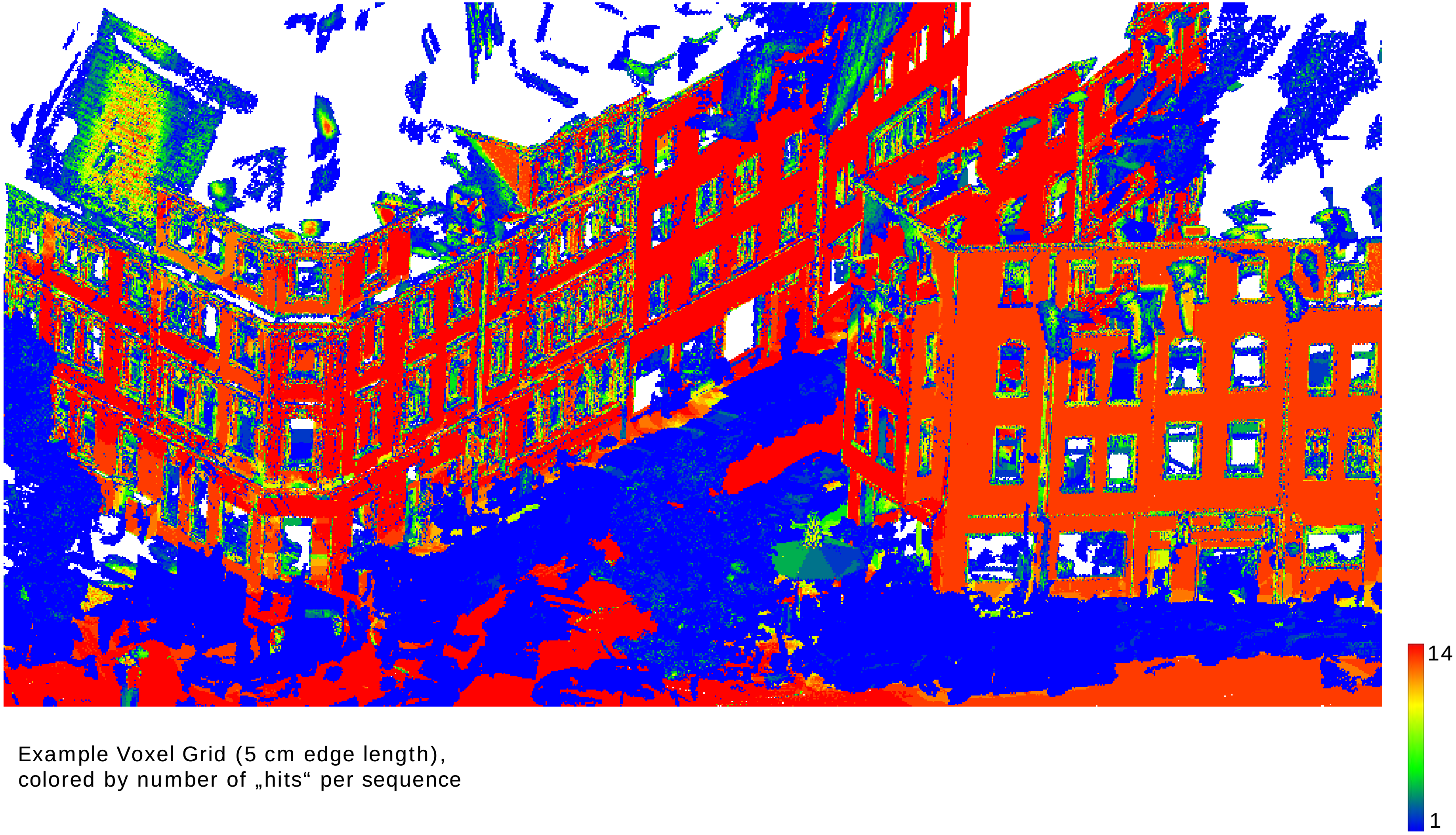
Observation Sequence

- Each voxel stores a sequence of observations:
  - For each measurement, a state is computed:
    - Hit: At least one reflecting point (surface)
    - Miss: At least one surface was traversed by a laser ray

- Example Sequence 2D

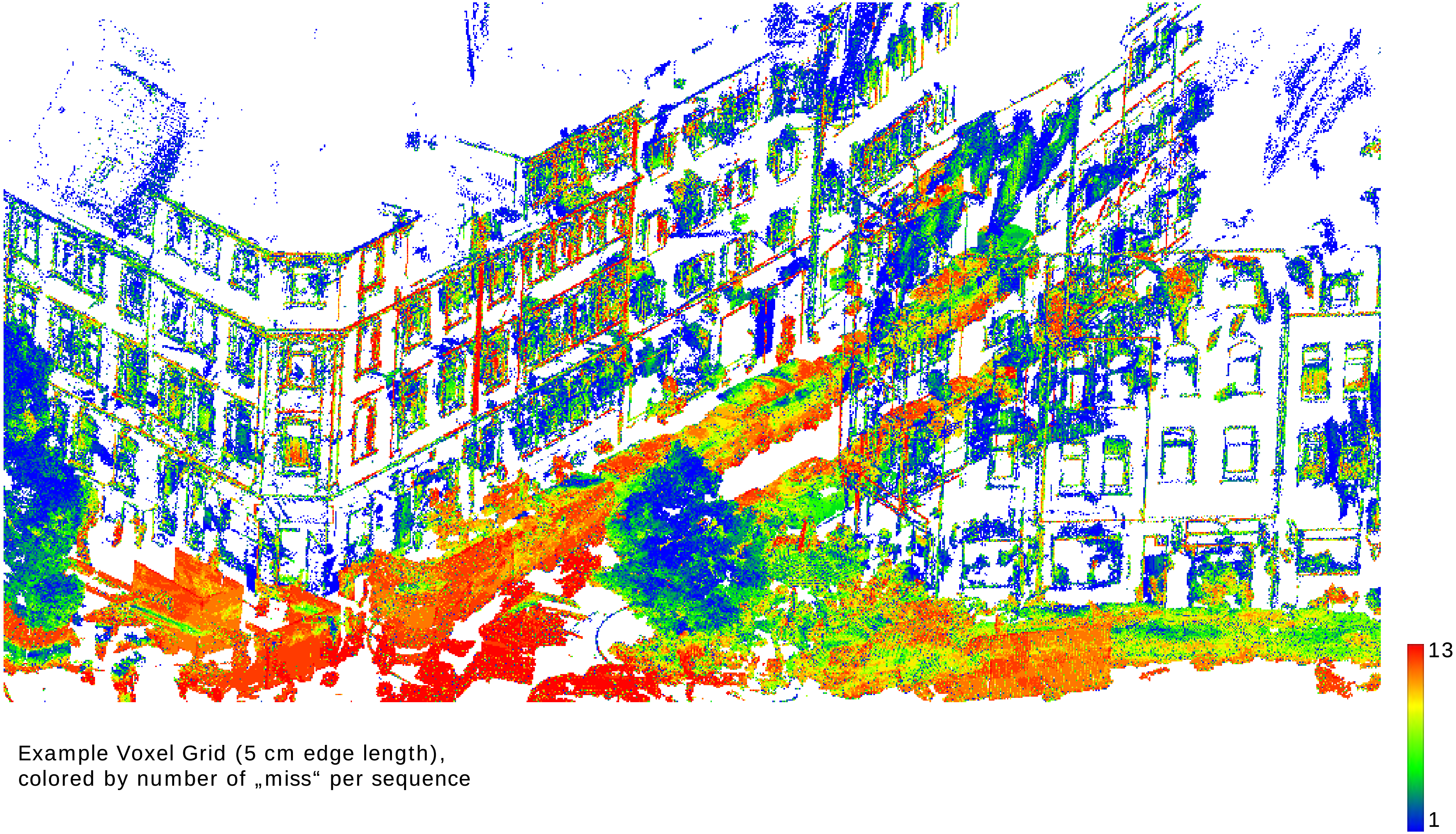
Measurement	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Hits	0	1	1	0	1	1	1	1	0	0	0	0	0	0
Misses	0	1	0	0	0	0	1	0	0	0	1	1	1	1

Voxel „Hit“ Count

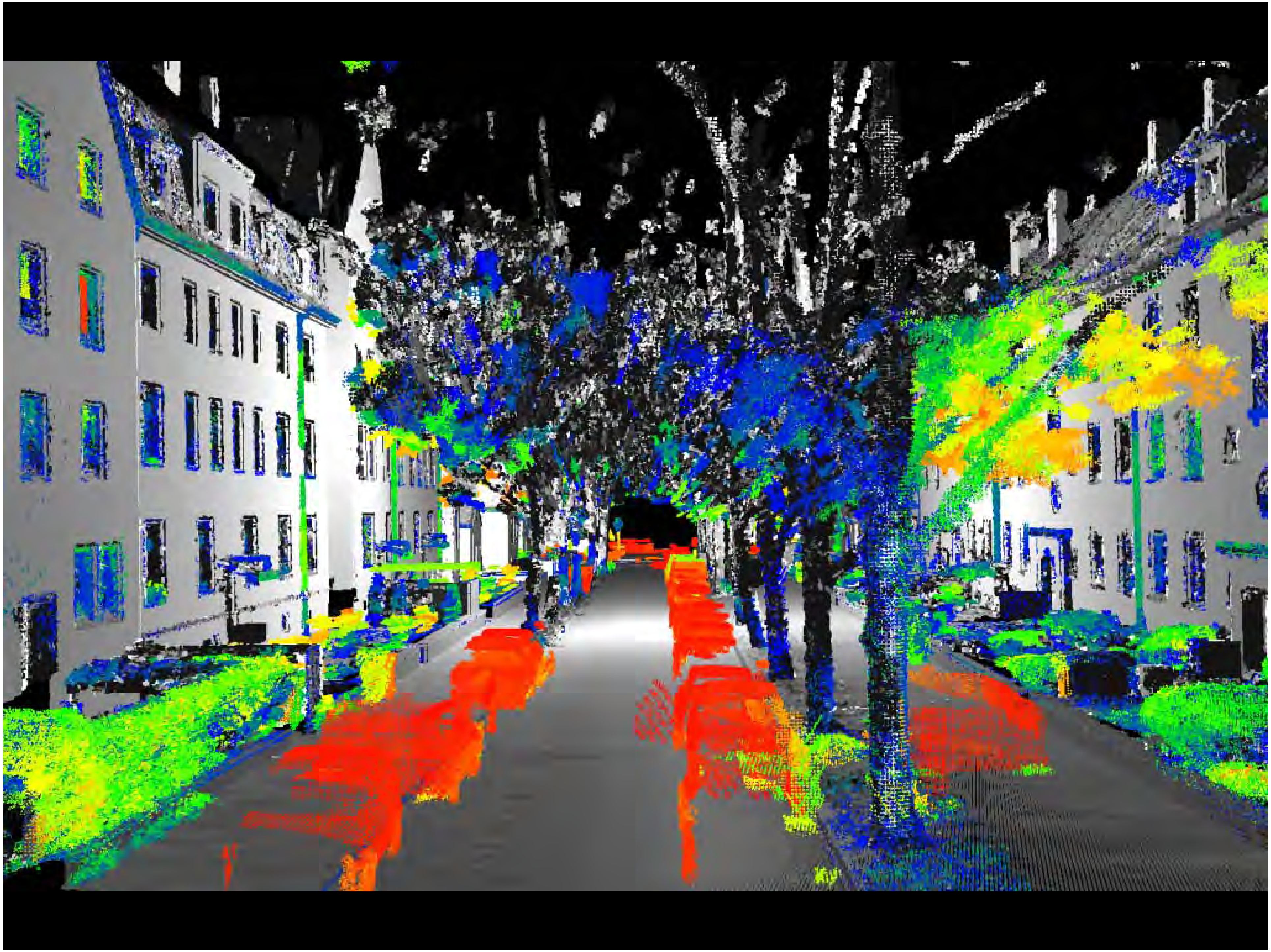


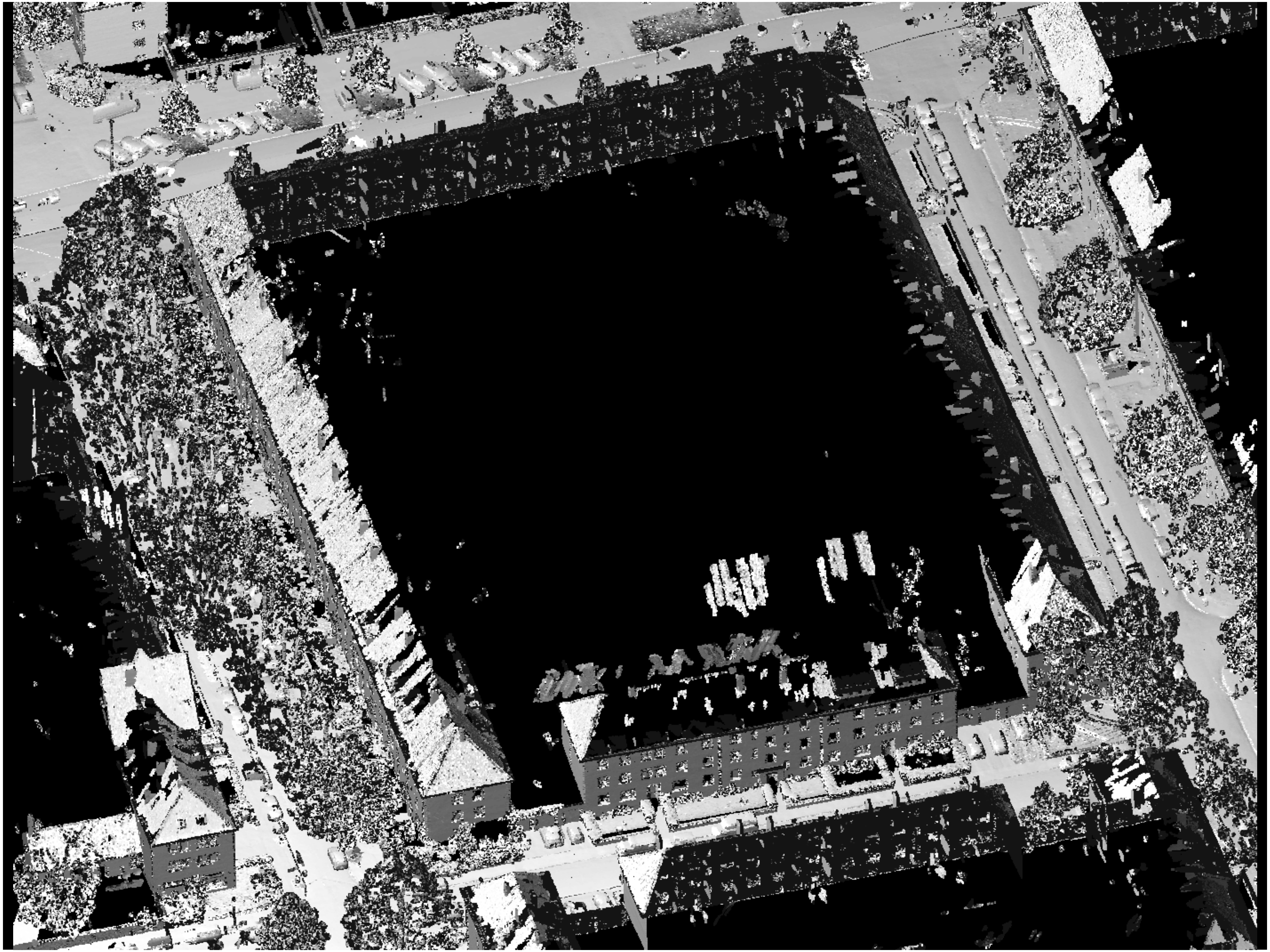
Example Voxel Grid (5 cm edge length),
colored by number of „hits“ per sequence

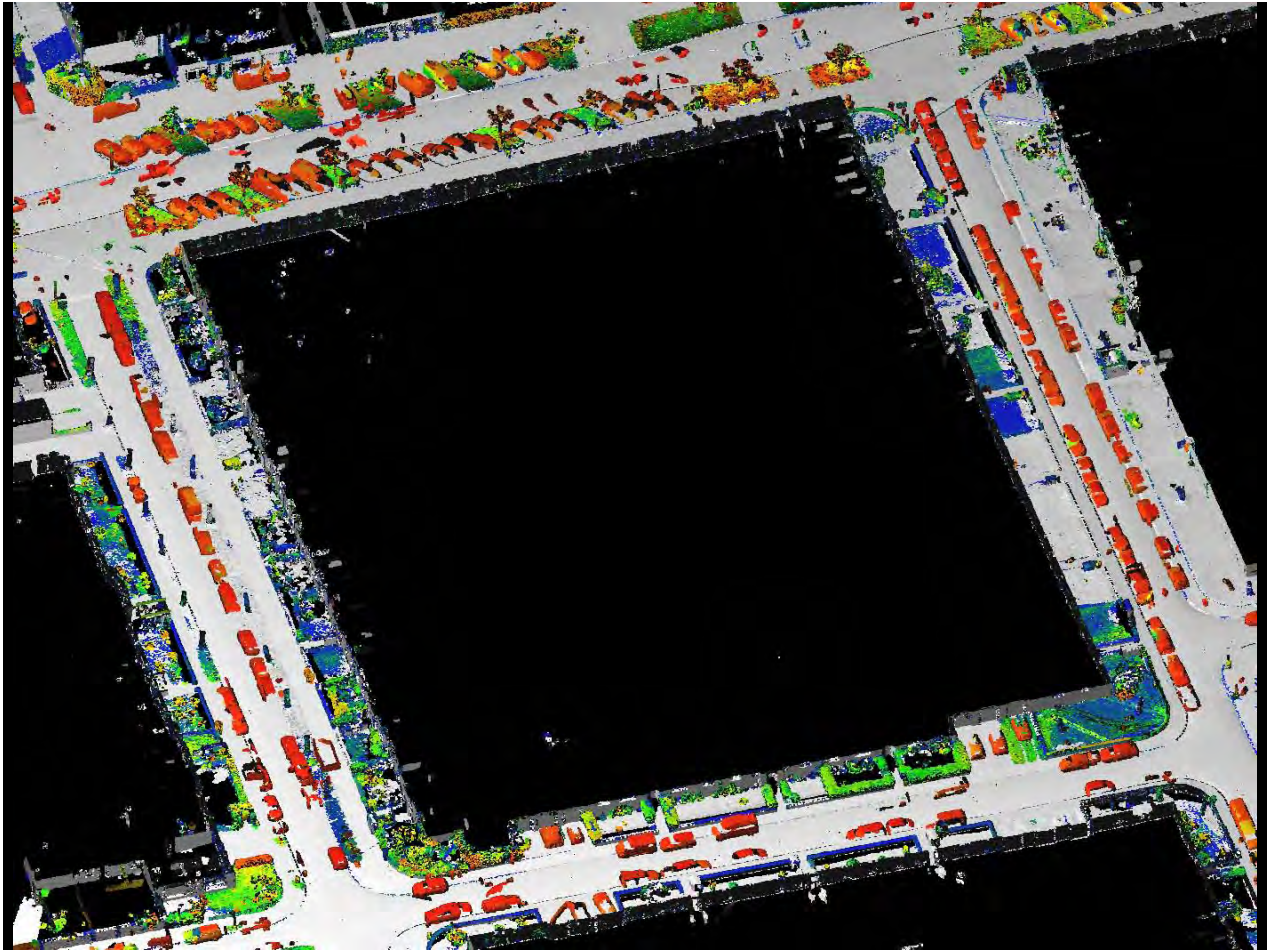
Voxel „Miss“ Count

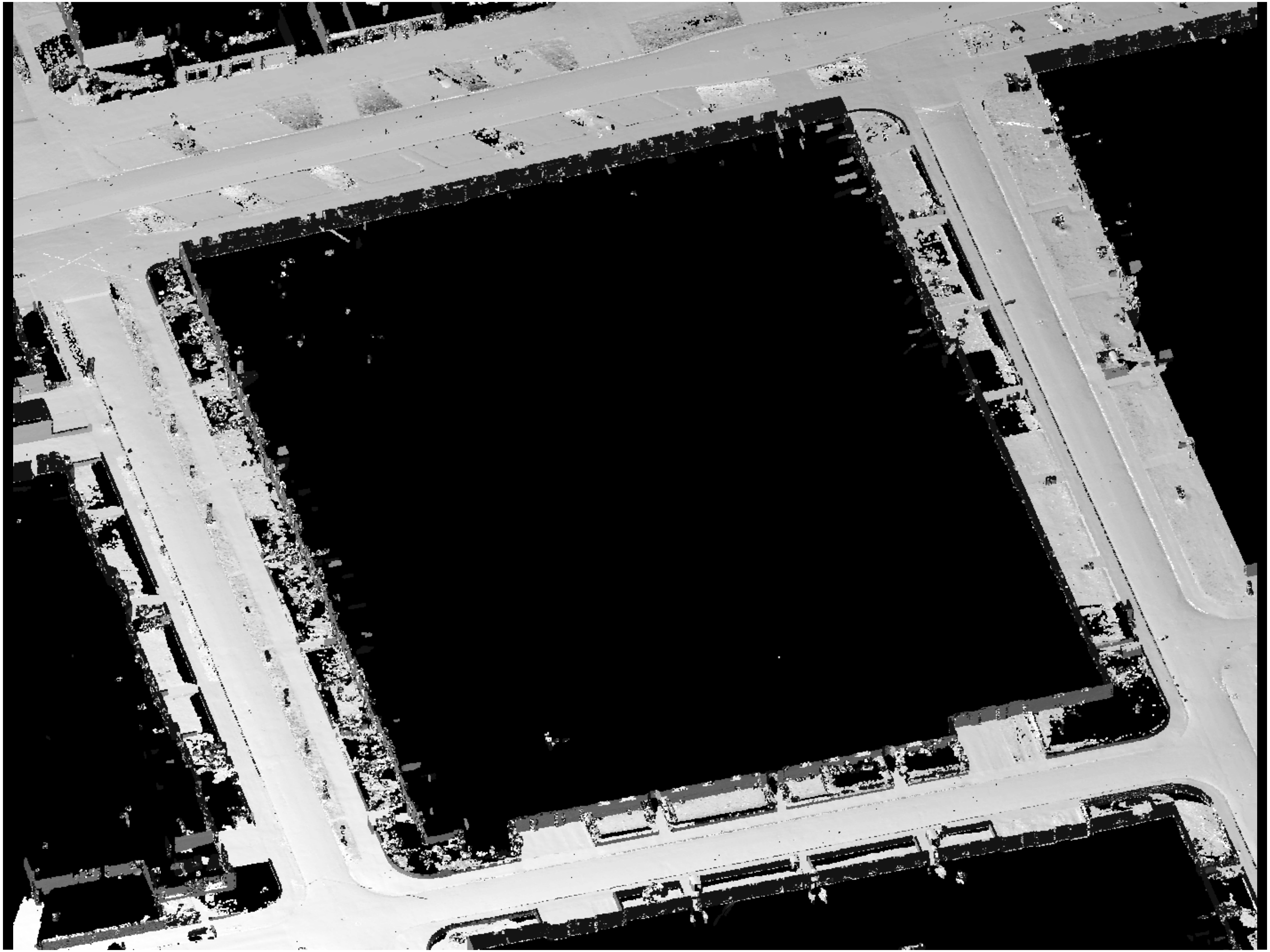


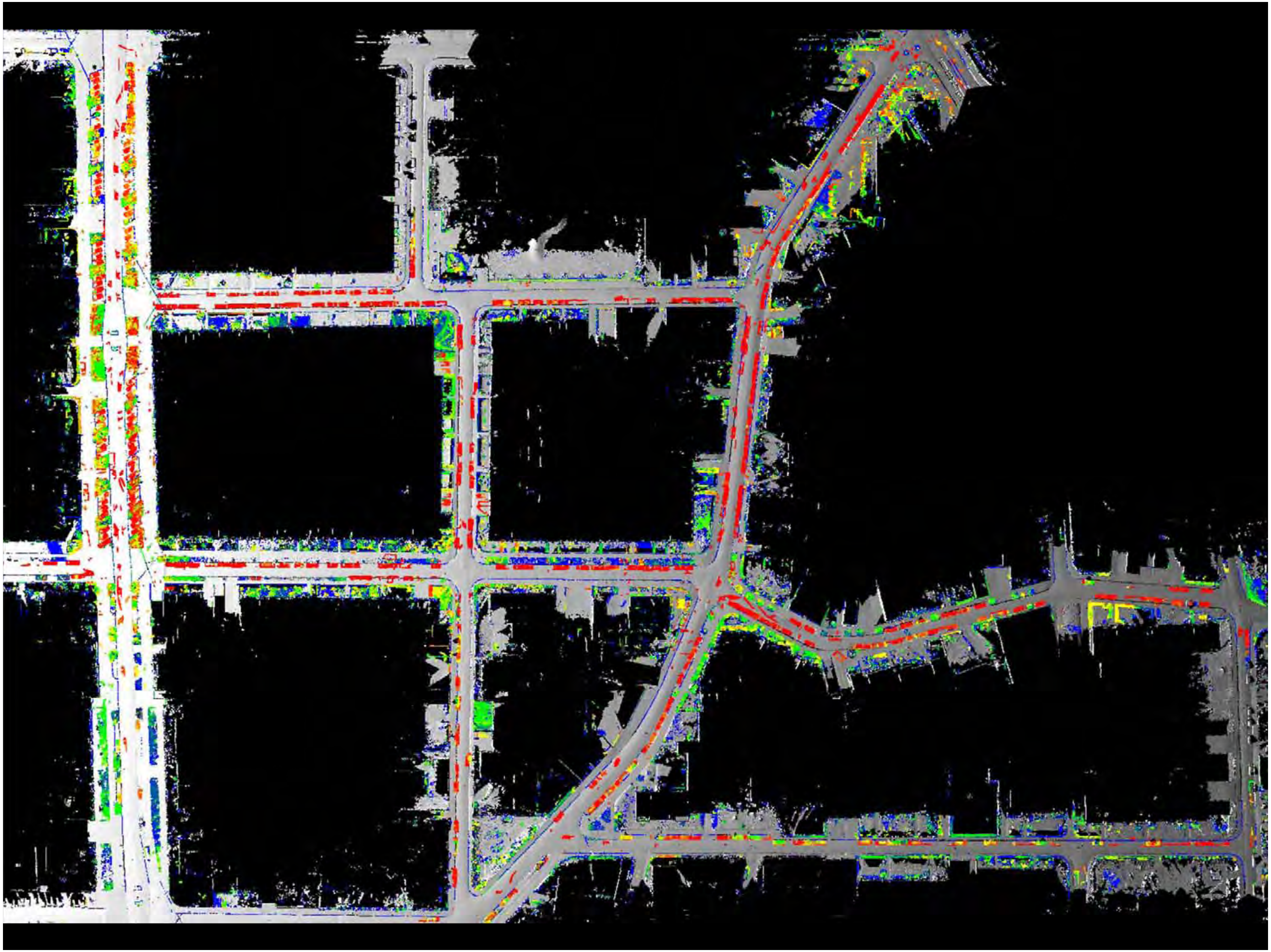
Example Voxel Grid (5 cm edge length),
colored by number of „miss“ per sequence

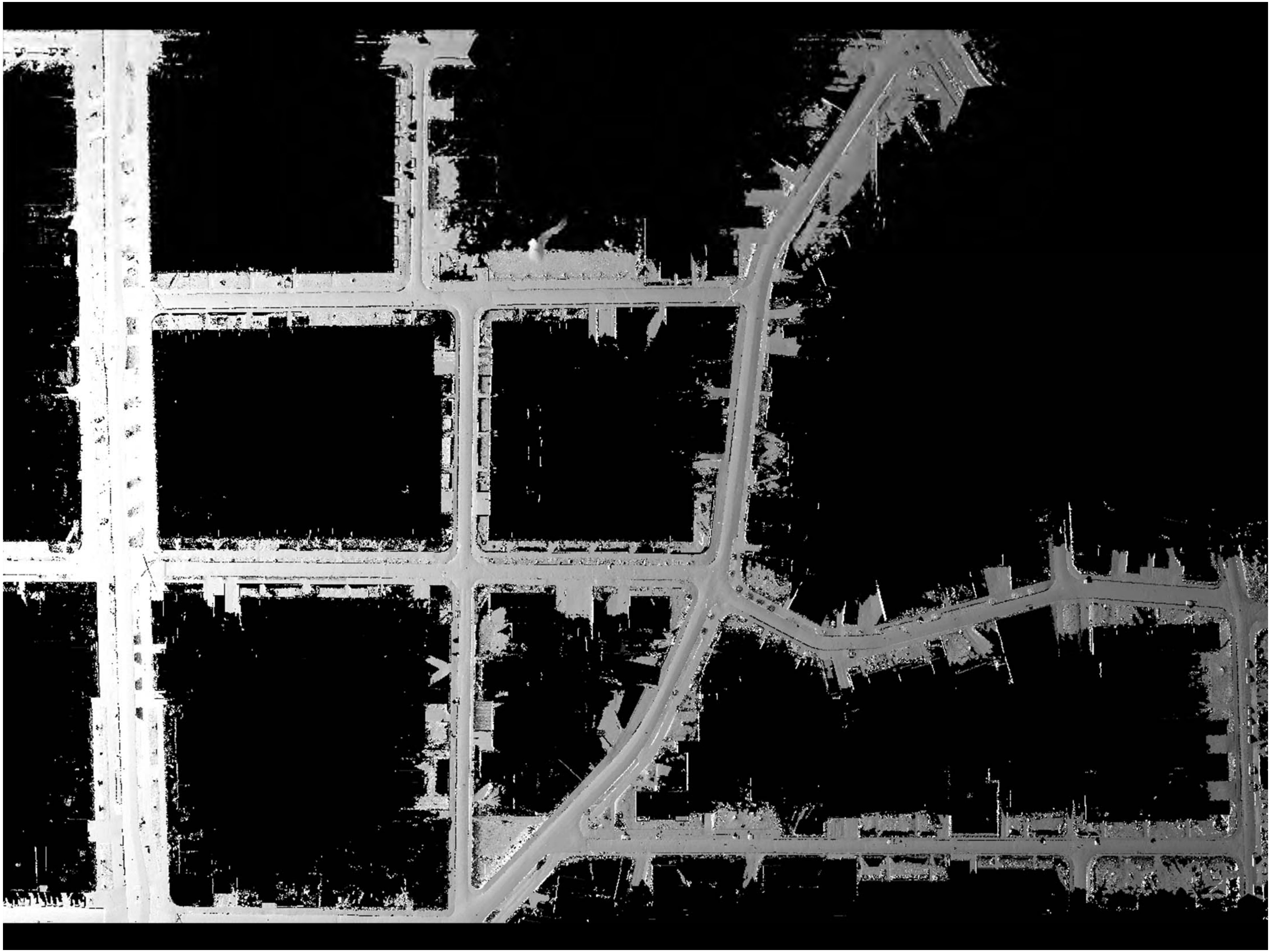






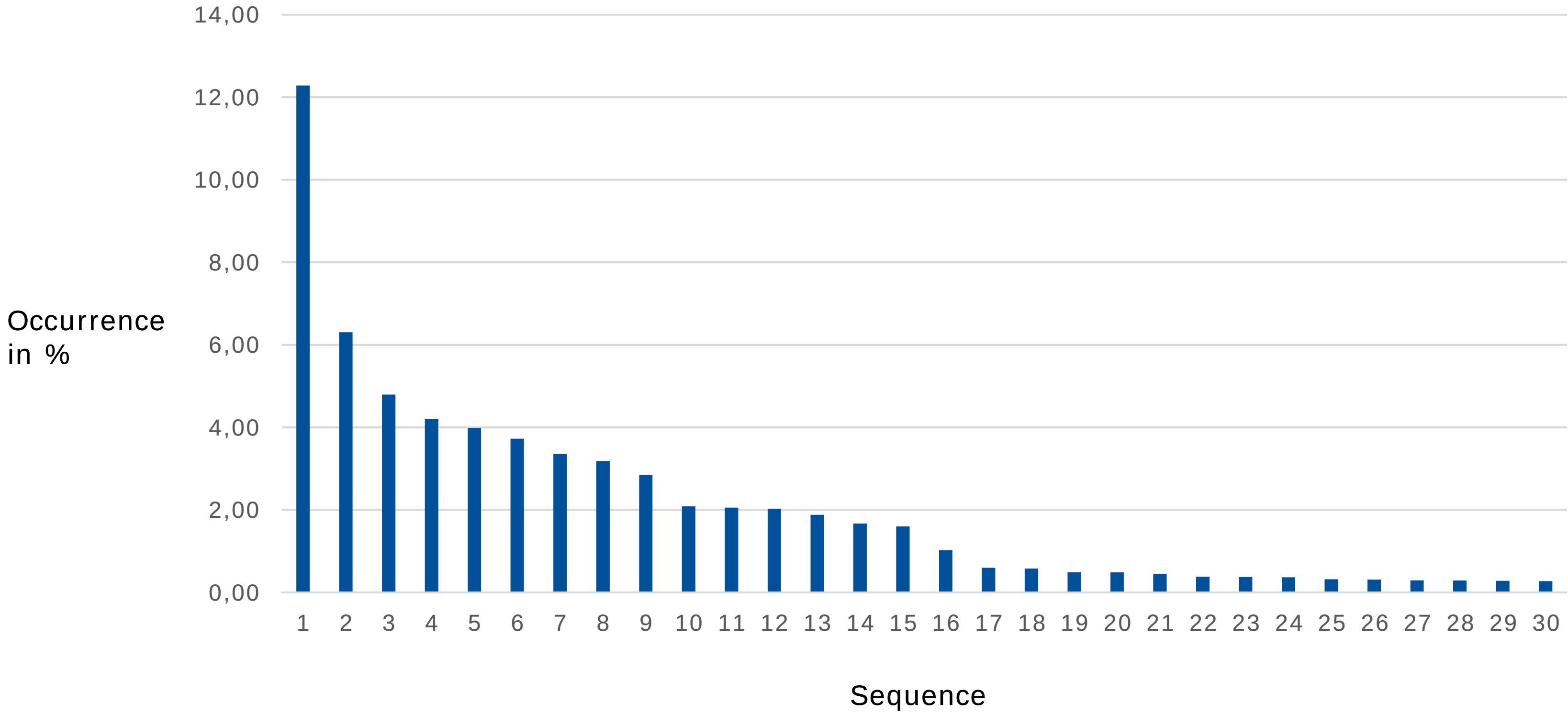






Histogram of „Hit"-Sequences (Julia Schachtschneider)

- $2^{14} - 1 = 16383$ distinct sequences (at least one hit)
- Most frequent sequences:



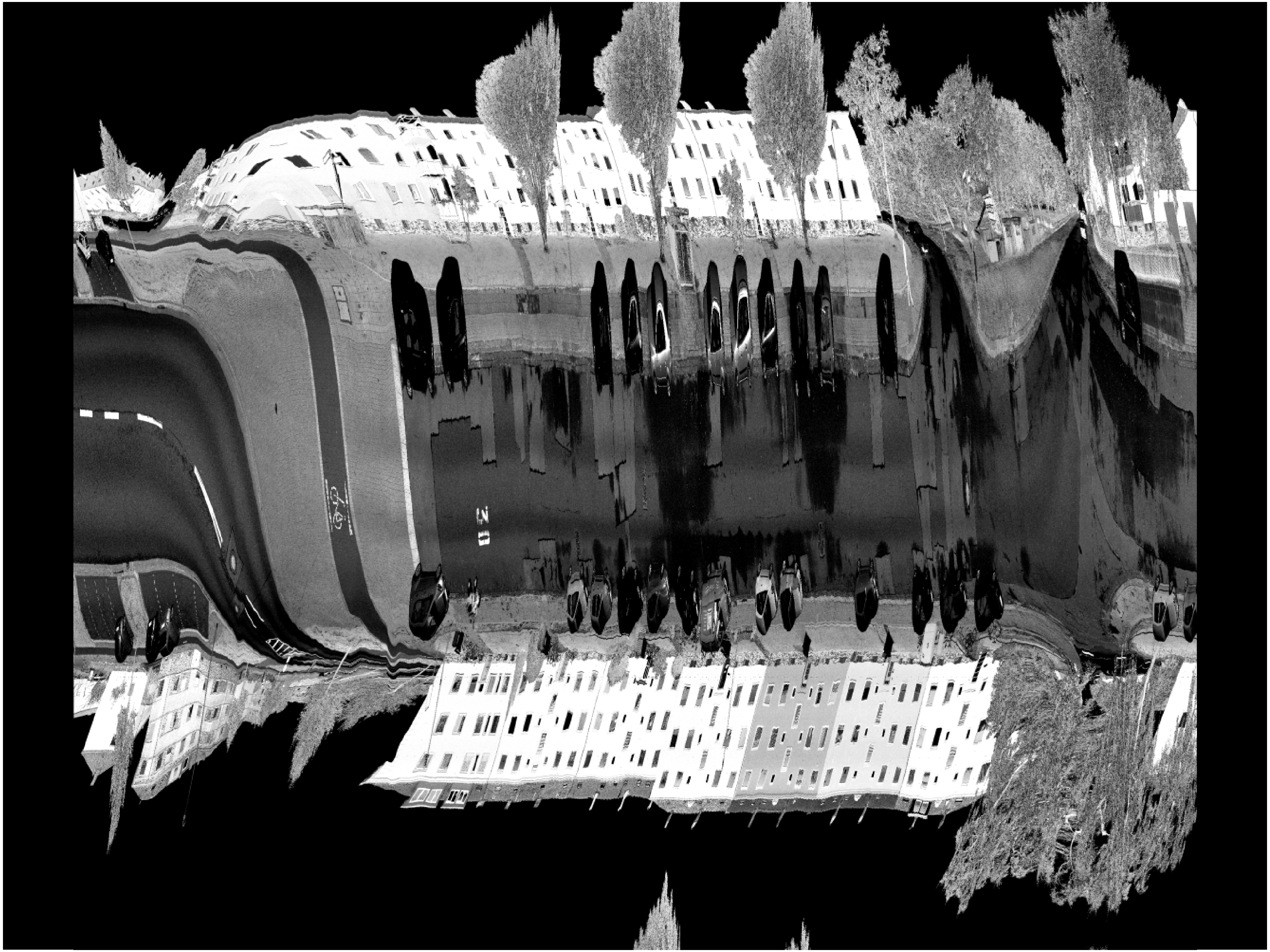
Most Frequent „Hit"-Sequences

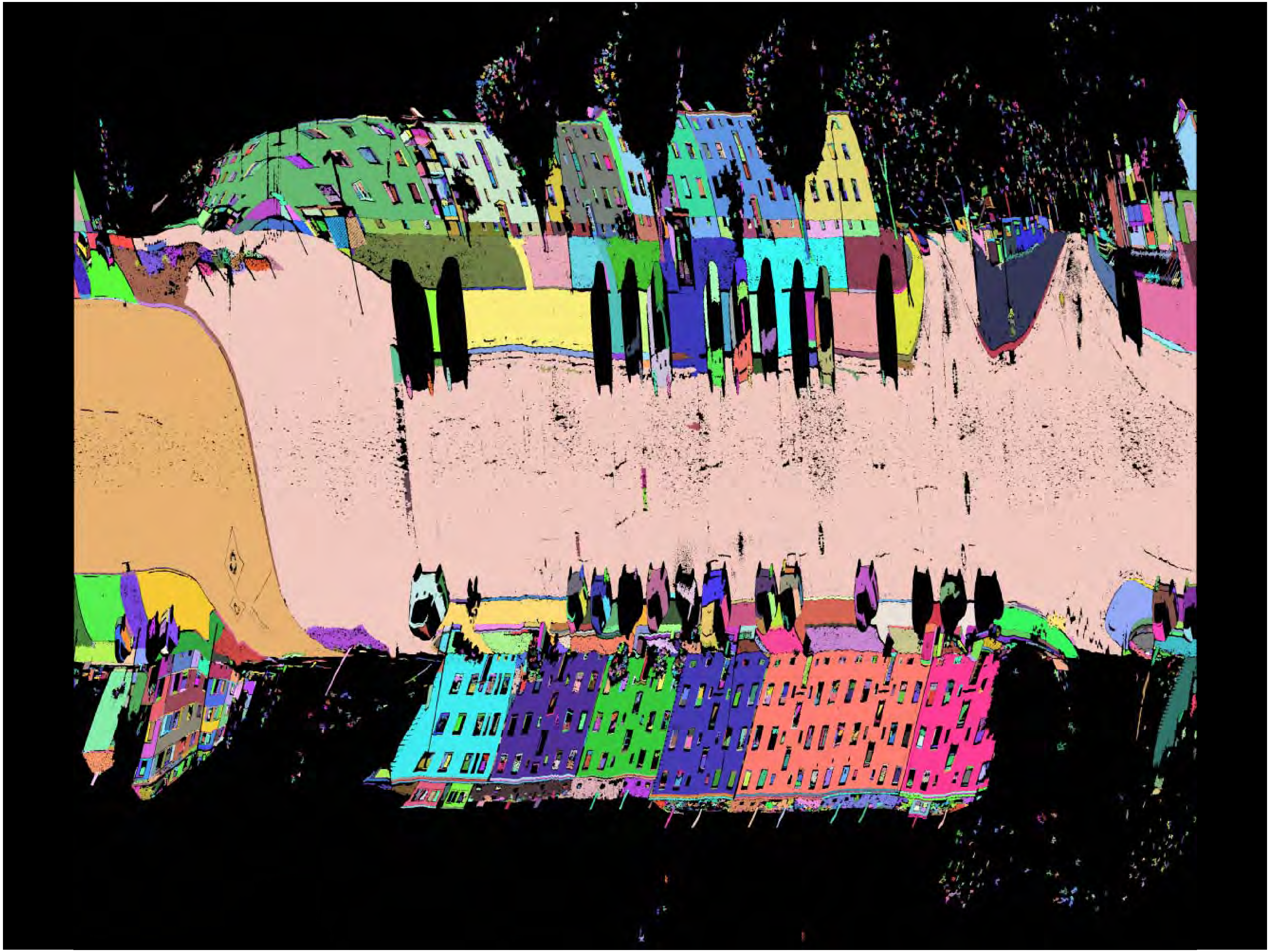
Sequence	170322	170328	170331	170405	170413	170428	170509	170606	170620	170704	170808	170823	170905	171004	%
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	12,28
2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	6,30
3	0	0	0	0	0	0	1	0	0	0	0	0	0	0	4,79
4	0	0	0	0	0	0	0	0	0	1	0	0	0	0	4,20
5	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3,98
6	0	0	0	0	0	0	0	1	0	0	0	0	0	0	3,72
7	0	0	0	0	0	0	0	0	1	0	0	0	0	0	3,36
8	0	0	0	0	0	0	0	0	0	0	1	0	0	0	3,18
9	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2,85
10	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2,09
11	0	0	0	0	0	1	0	0	0	0	0	0	0	0	2,06
12	0	1	0	0	0	0	0	0	0	0	0	0	0	0	2,03
13	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1,88
14	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1,67
15	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1,60
16	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1,02
17	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0,60
18	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0,58
19	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0,49
20	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0,48

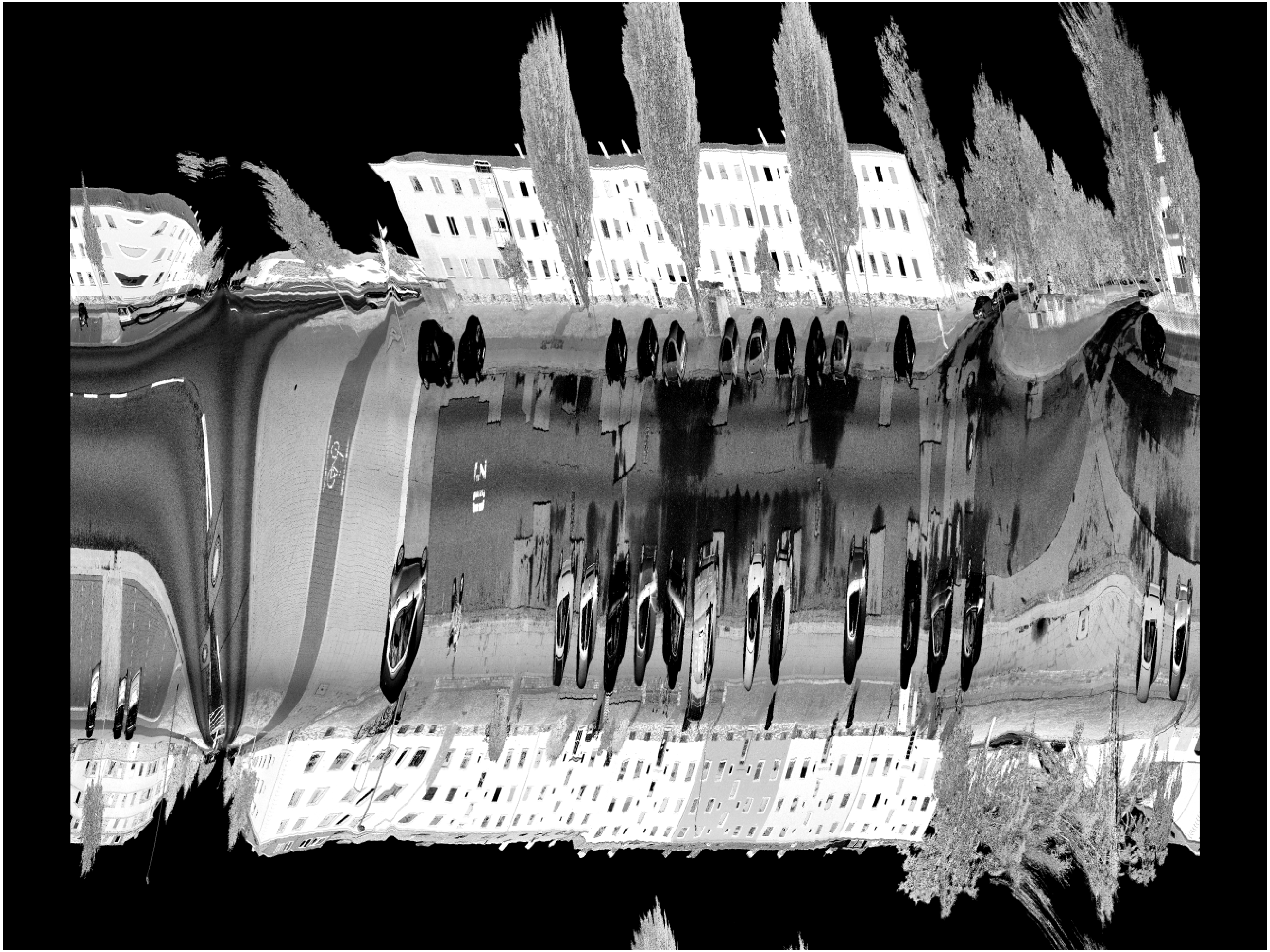
Interpretation: Segmentation based on geometry only

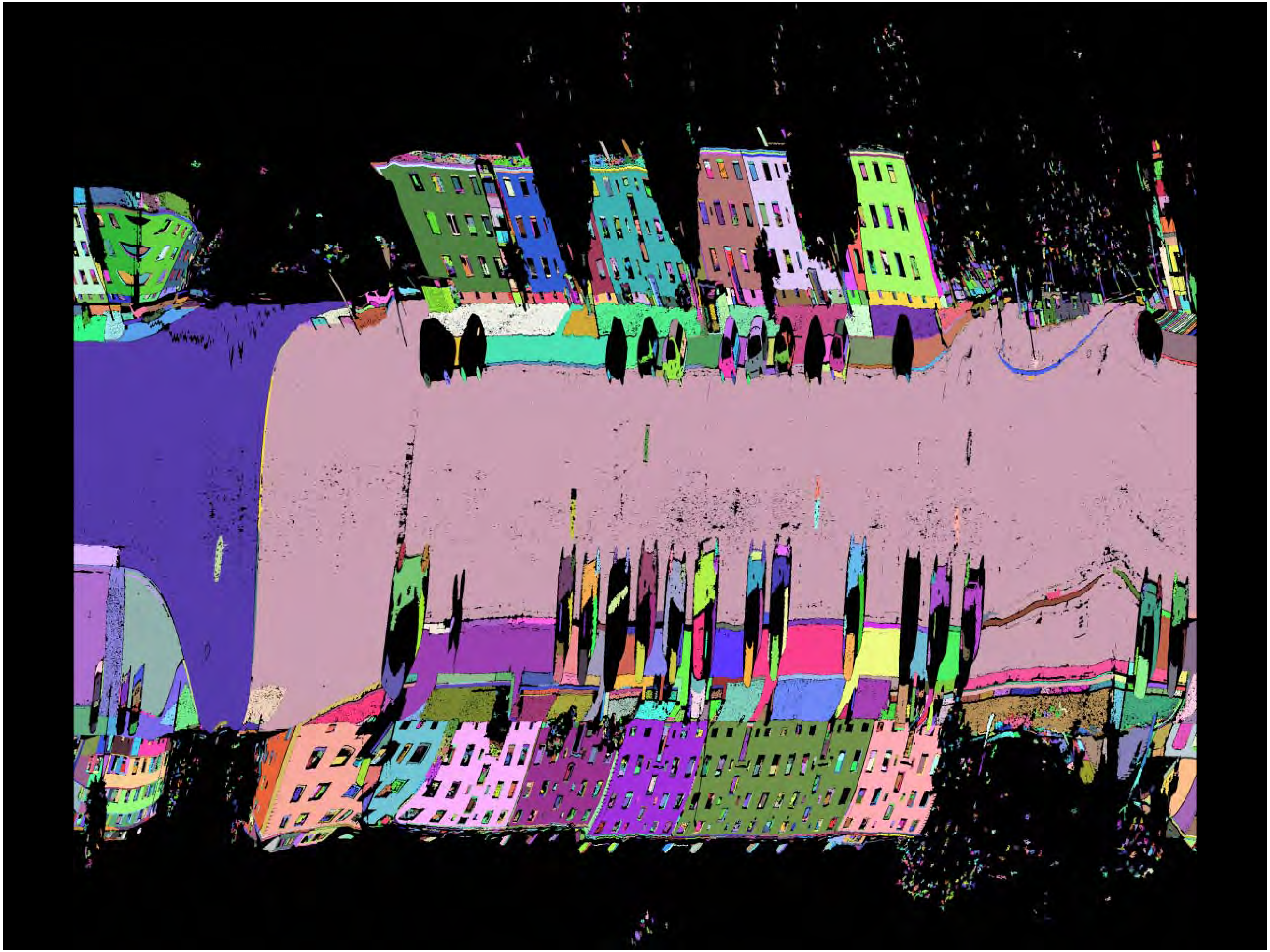
Segmentation of scan strips

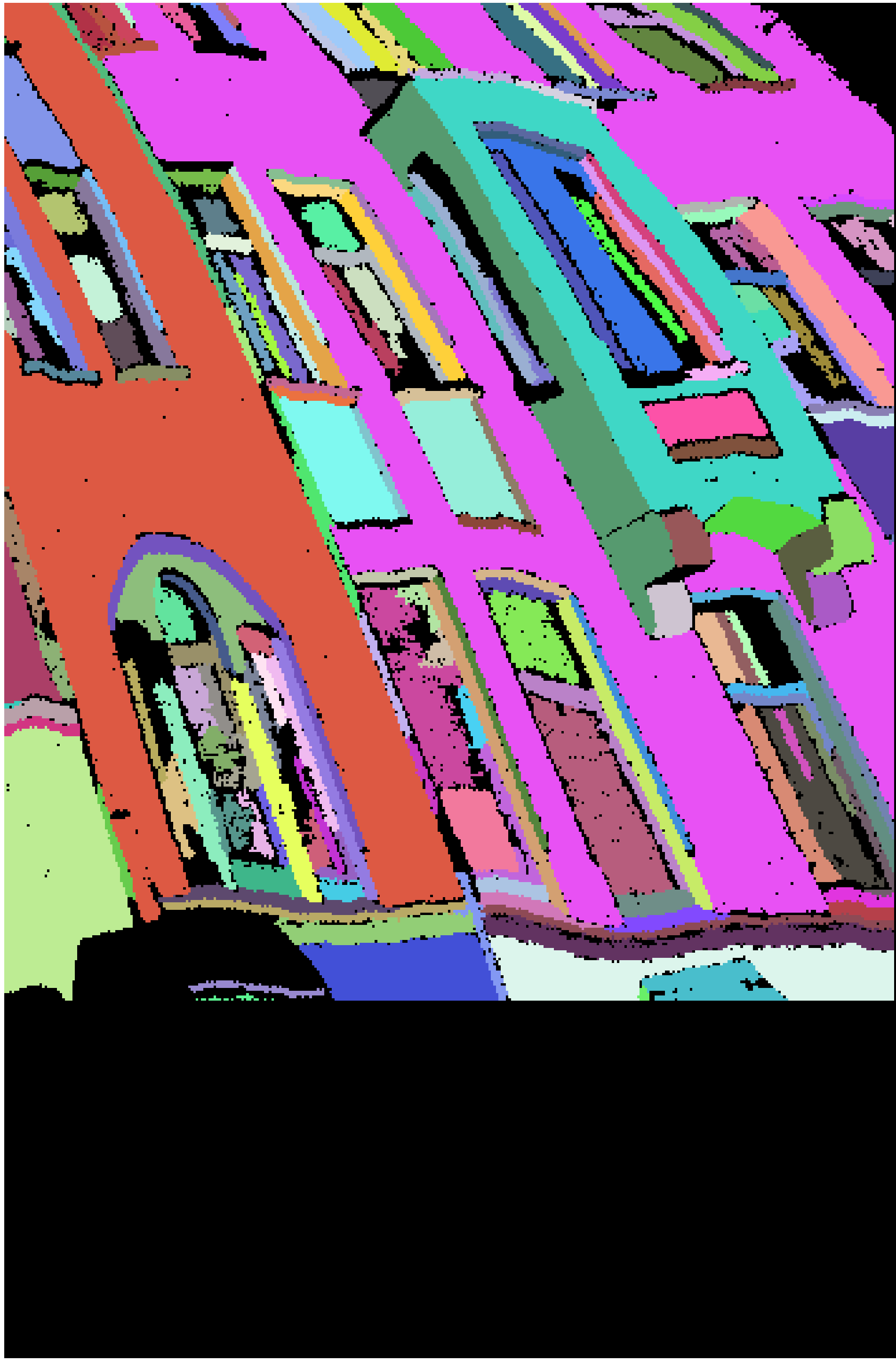
- Efficient graph-based segmentation
- Combines “image topology” and “object space geometry”
  - Processing operates on scan strips
- Homogeneity criterion: geometric C^0 and C^1 continuity

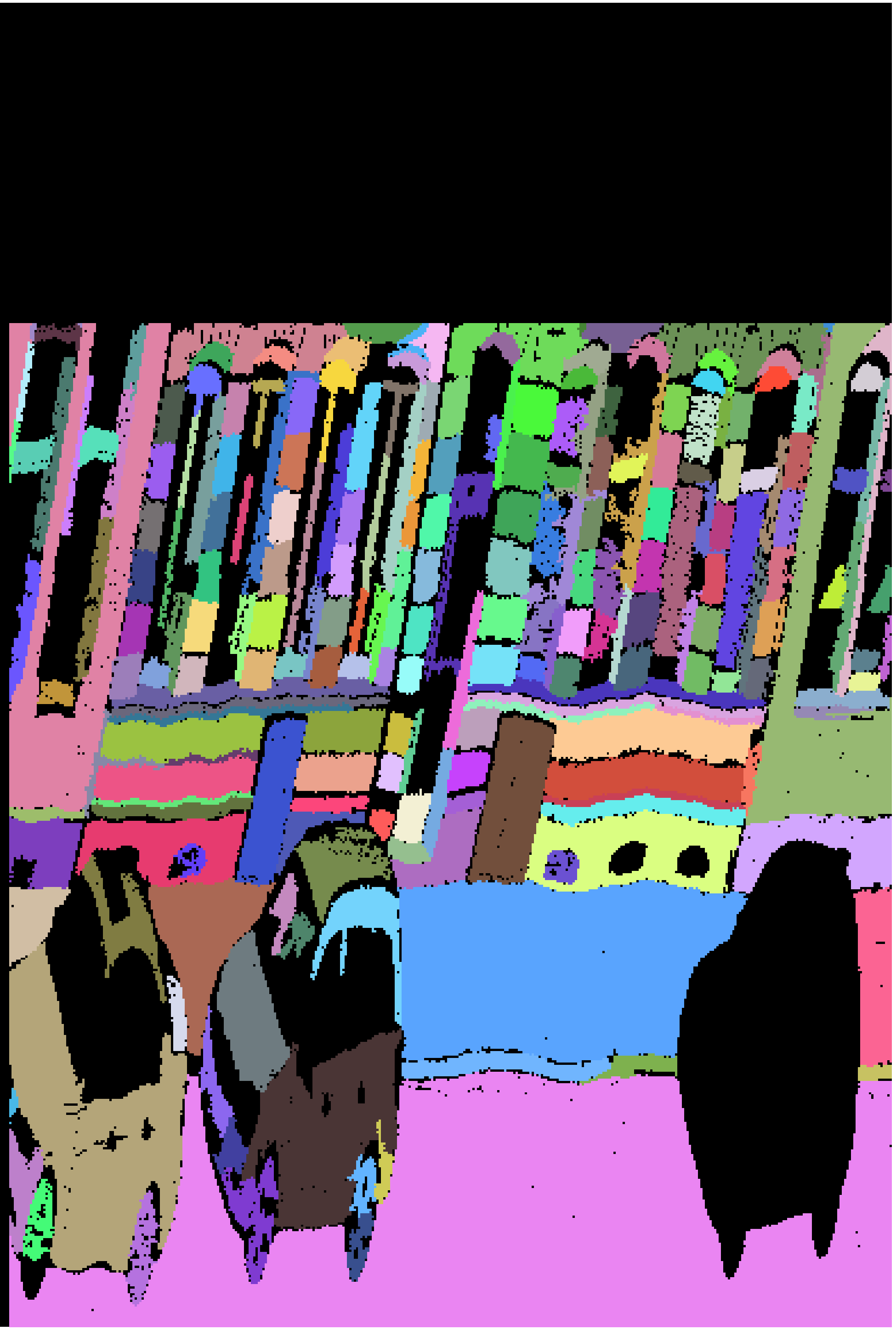










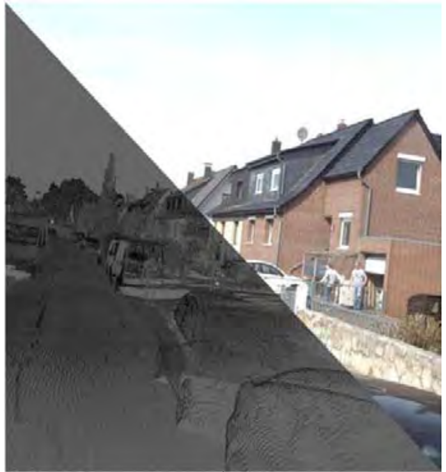


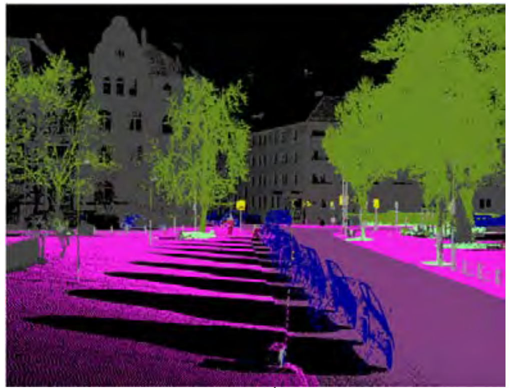
Interpretation:
Point cloud classification using label transfer

Torben Peters @ ikg

Motivation

Cityscapes dataset



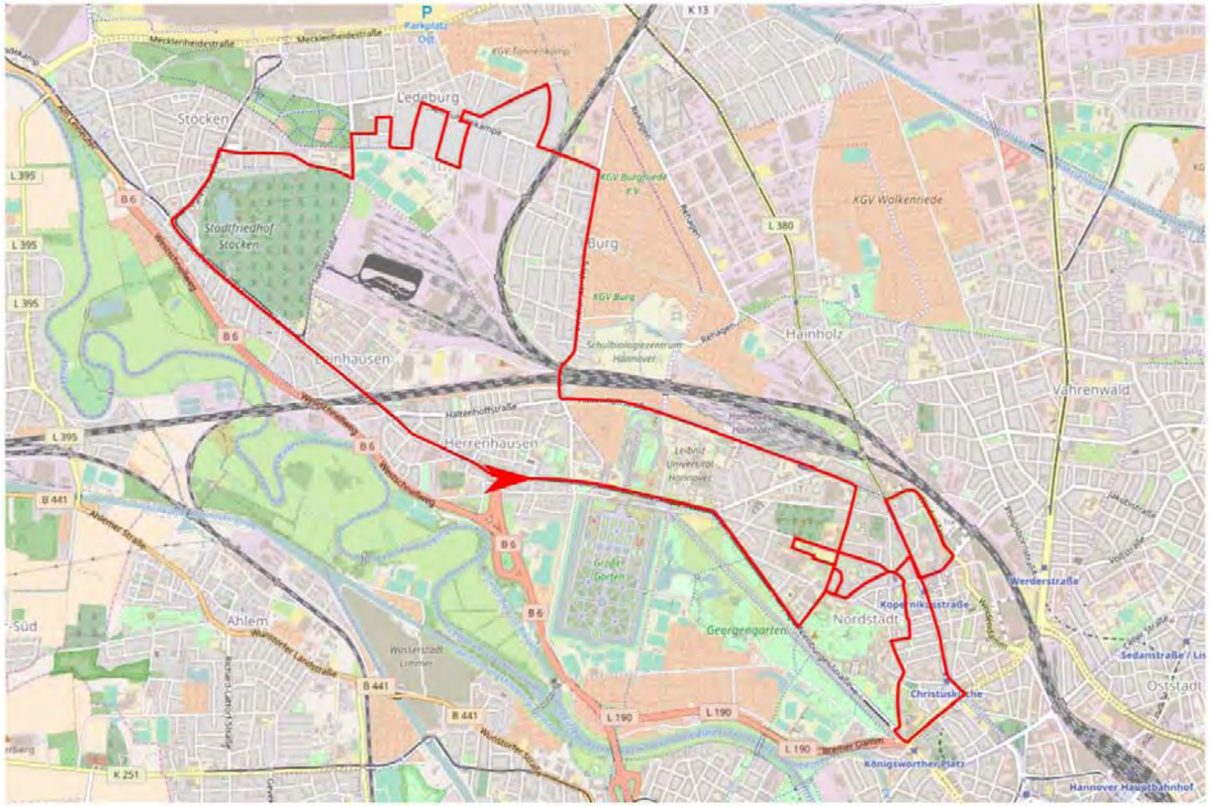


Our data set: point cloud fused with images

Annotated 3D point cloud

Our data set

- The long-term, biweekly mobile mapping data acquisition
  - Resulting in over **24 billion 3D points**
  - And over **250.000 images**



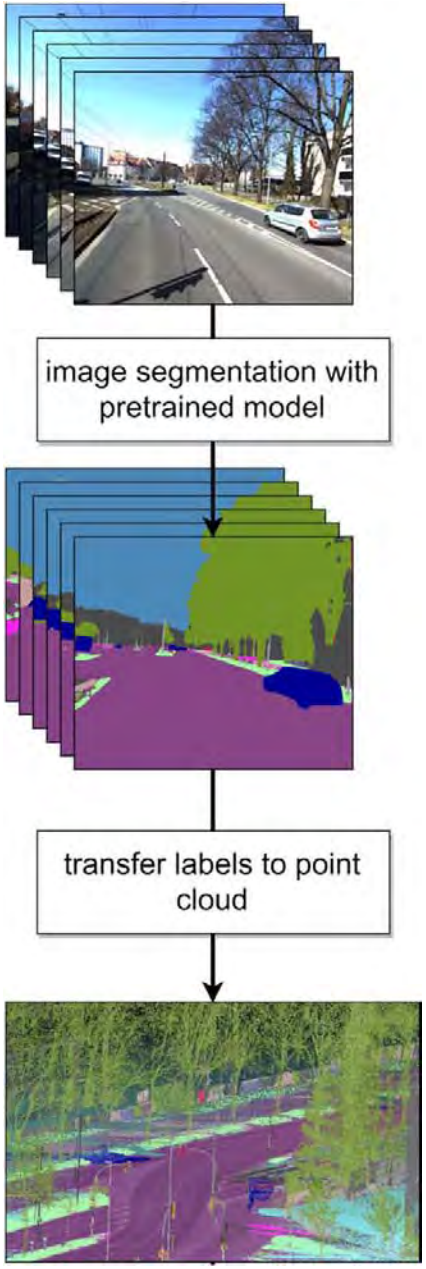
Cityscapes data set

- Consists of 25.000 annotated images
- 50 cities
- 20-30 classes
  - Street, Sidewalk, Person, Car…
- Best classification solutions are based on Deep Learning

Source: Cityscapes

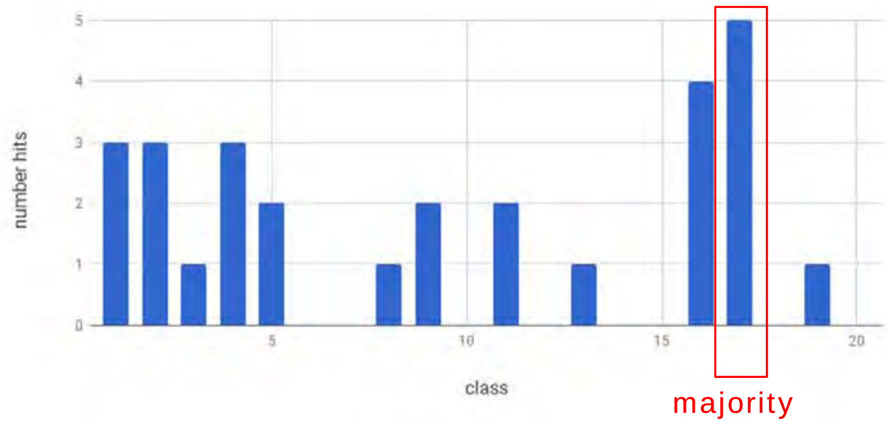
Label transfer

- During our measurement campaign we captured images with 1 Hz
- These images are semantically segmented using PSPNet
  - PSPNet was pre-trained on the Cityscapes data set
- The labels are mapped to the point cloud by projecting the 3D points to the image planes



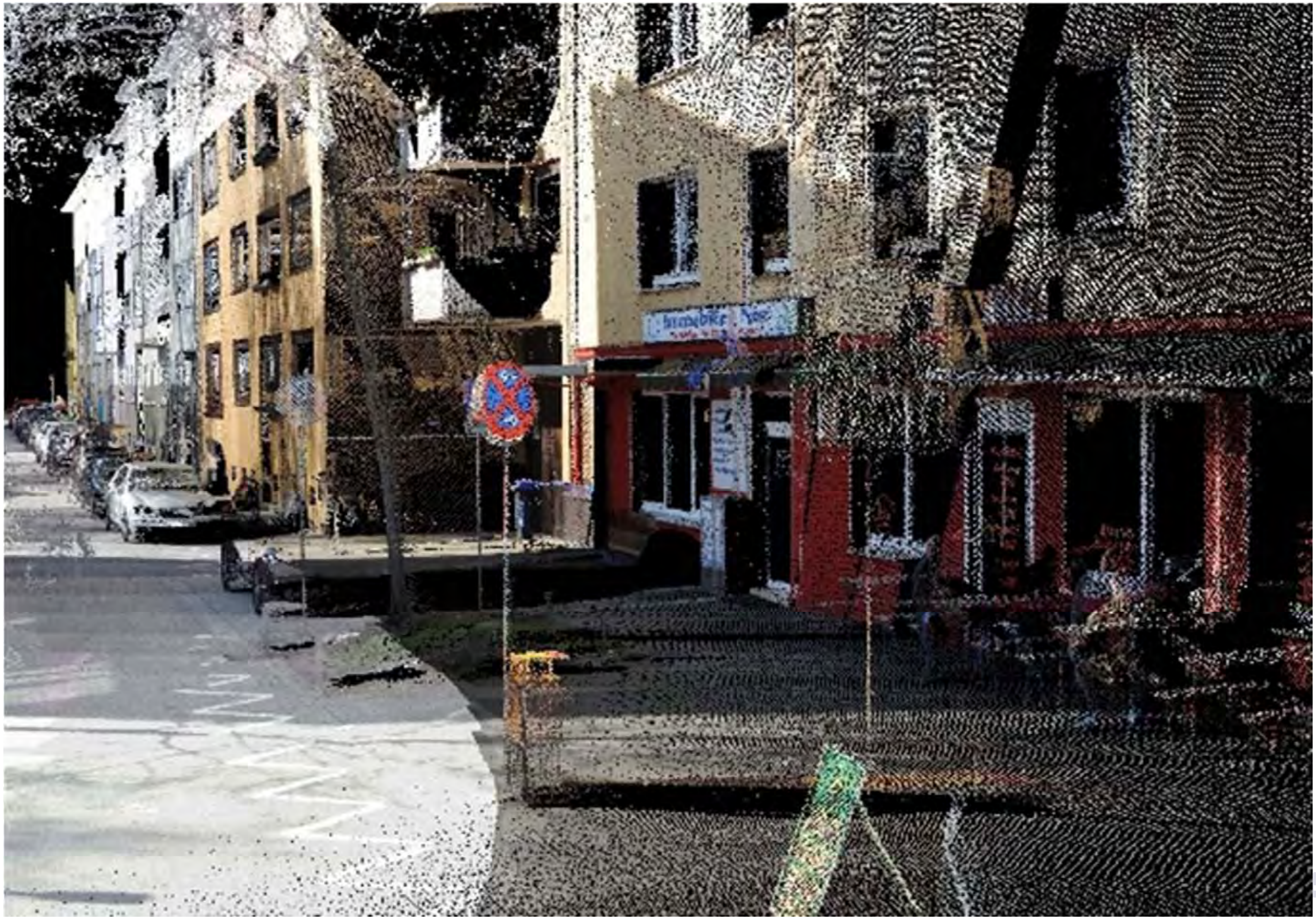
Label transfer

- Problem:
  - For nearly every laser point we are capturing more than one image pixel
  - Therefore we are aggregating all transferred labels in one histogram per point
  - Histogram: class vs. occurrence in images



- The resulting histogram can contain contradictory information due to different reasons

Label transfer



kg

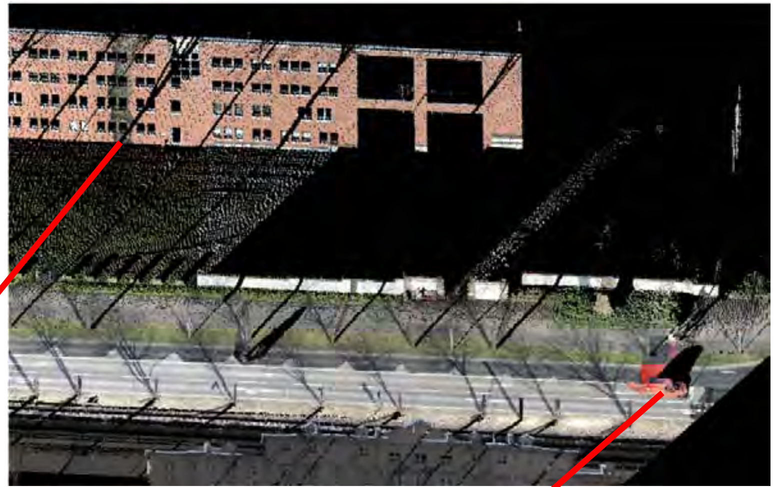
Label transfer

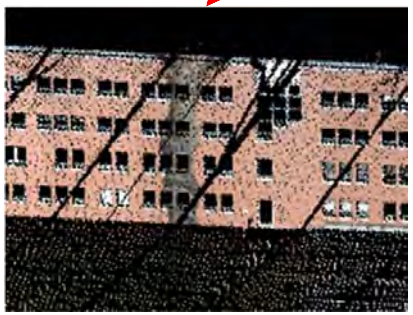
kg

Label noise: occlusions

- **Noise due to occlusion**
  - laser and image rays do not coincide
  - 3D points being assigned the label of an occluding object
- This effect is mitigated to a certain extend by the accumulation of labels in histograms

point cloud colored without ray tracing





Building colored with tree trunk pixels

Road surface colored with car pixels

Label noise: occlusions

- Further improvement by doing **ray tracing**
  - We used a 10 cm voxel grid
  - When determining the label of a 3D point, the ray to each camera center is traced in this grid
  - The point is considered to be occluded if an occupied cell is found along the ray

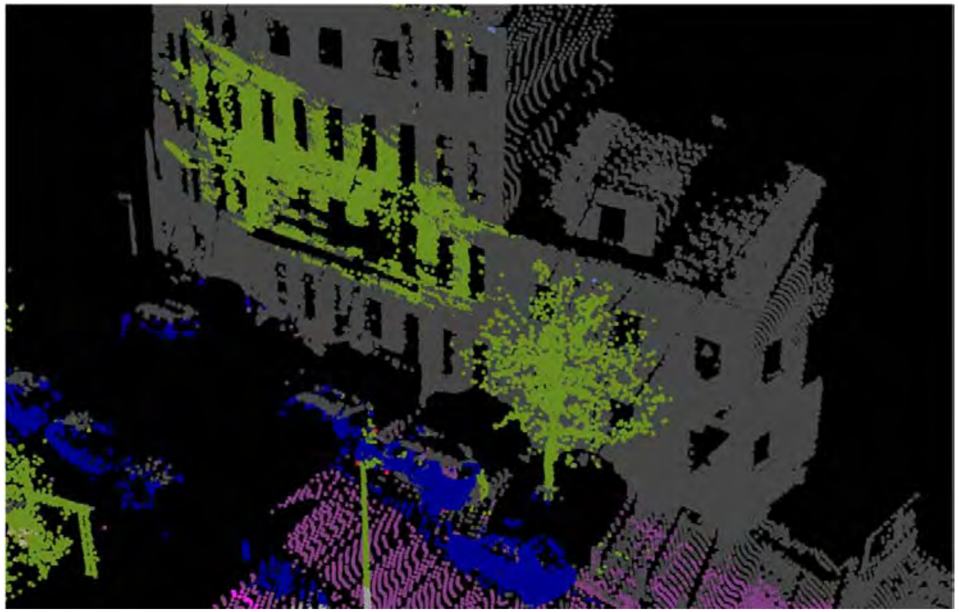
point cloud colored without ray tracing

point cloud colored with ray tracing

Label noise: label policy

- **Noise due to label policy**:
  - Surfaces behind tree crowns have to be assigned the tree label
  - Some rays are going through tree crowns
    - Some occluded points like facades will be labelled as tree

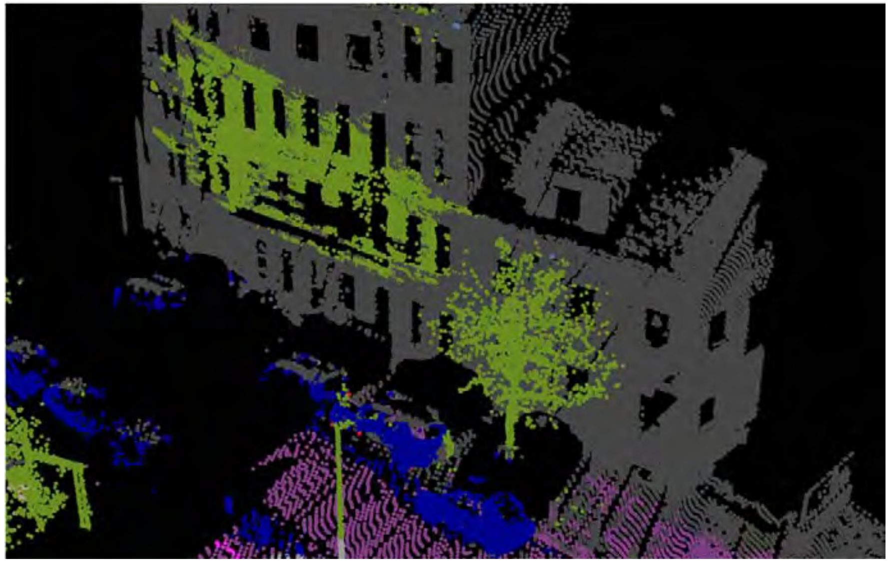
Semantically segmented image



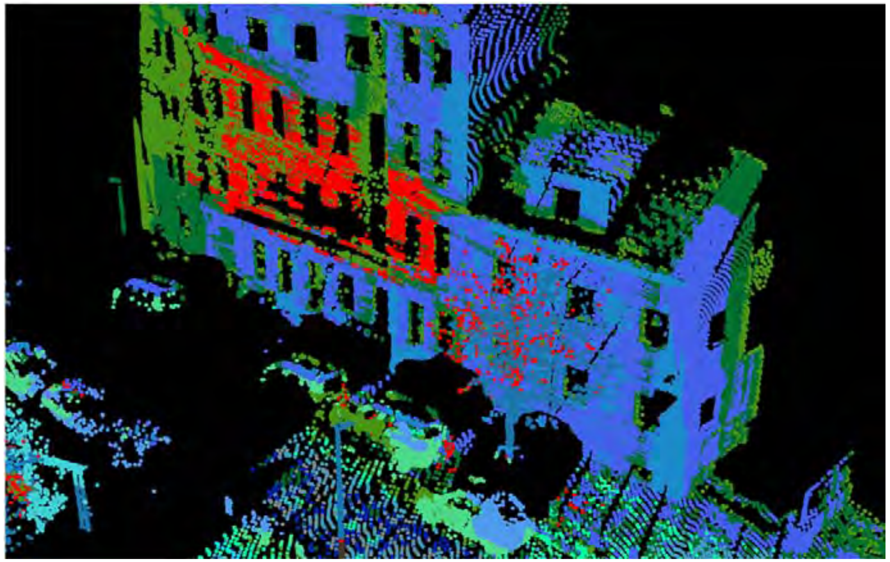
Point cloud colored by majority label

Label noise: label policy

- By storing the aggregated labels in a histogram we are also accumulating labels from the object in the front
  - We can use this information in order to identify the noisy histograms
- Example shows that k-means clustering is able to separate this kind of noise
  - The histogram can be used as a feature vector



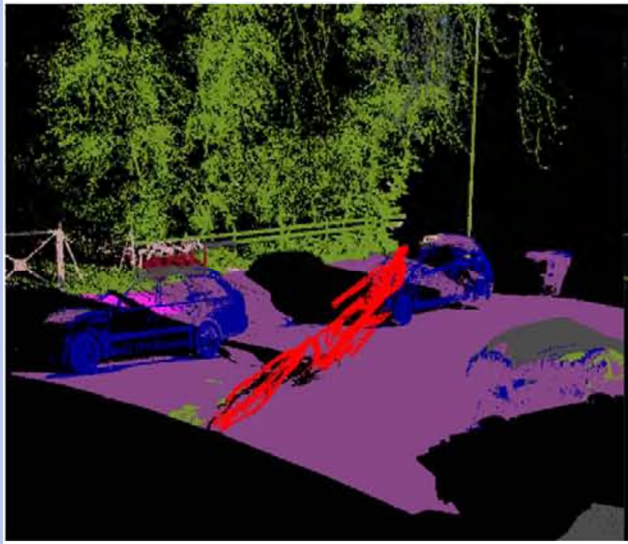
Colored by major label



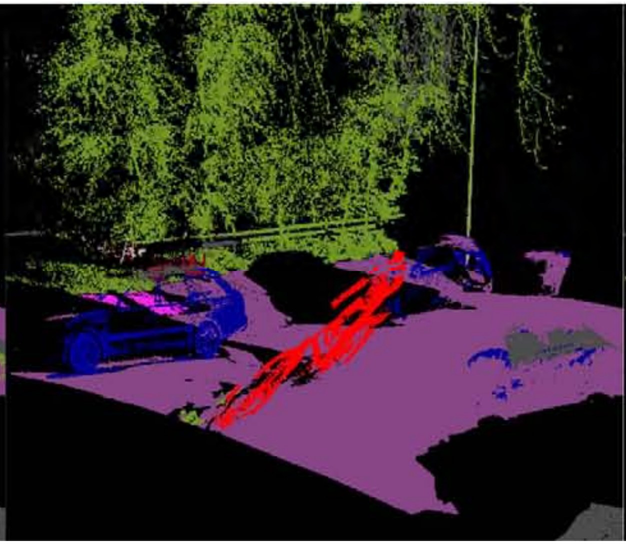
Clustered by k-means: clusters shown using random colors

Label noise: dynamic scenes

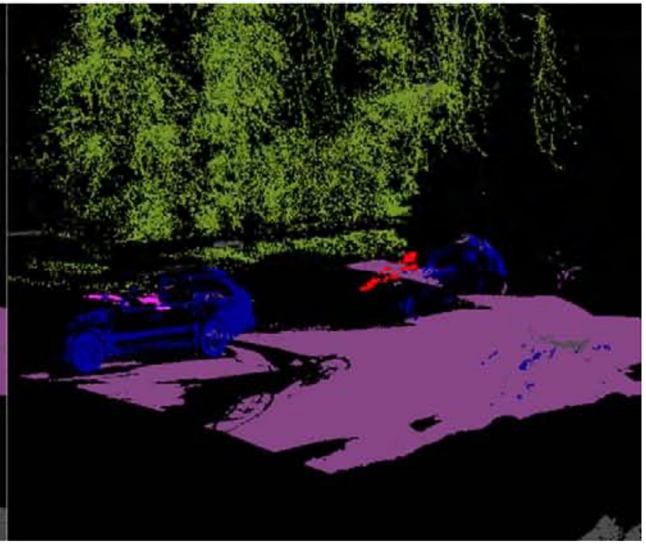
- **Noise due to difference in capture time**
  - Use histogram, voxelization and campaign count to reduce the error



Unchanged dataset



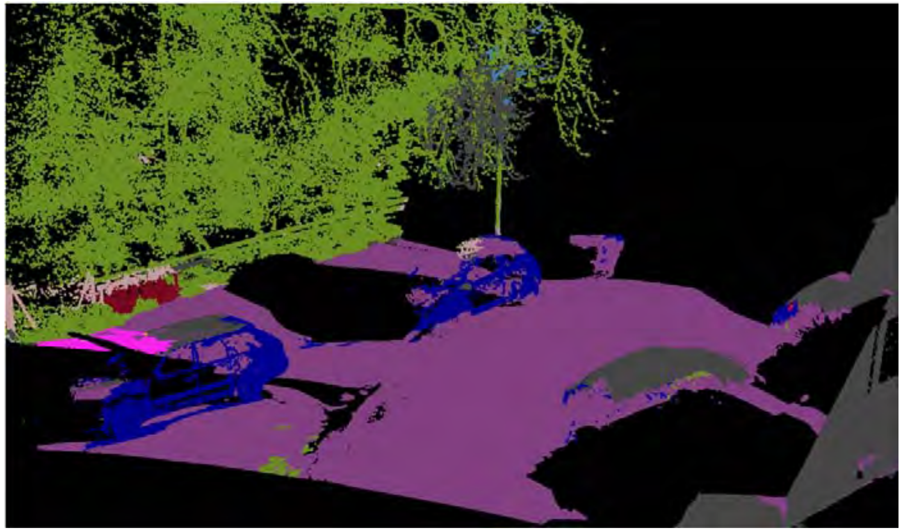
Cleaned dataset using only the histogram as feature



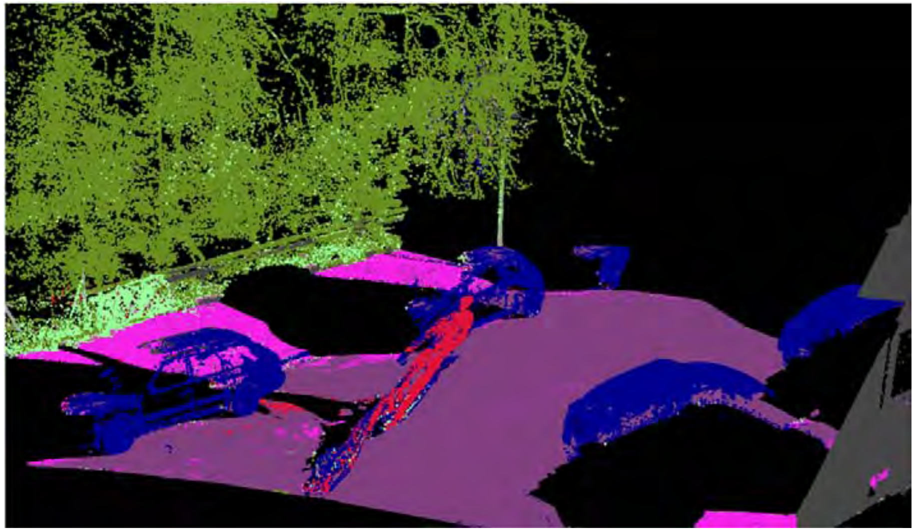
Cleaned dataset using also the campaign count

Reduction of label noise

- We were able to rise the IoU to about 63.79% which is only 18% worse than PSPNet
- Using this approach we were able to recover erroneous labeled 3D points e.g.
  - Sidewalk appears
  - Bicycle rider is mostly labeled correctly
  - Cars are corrected



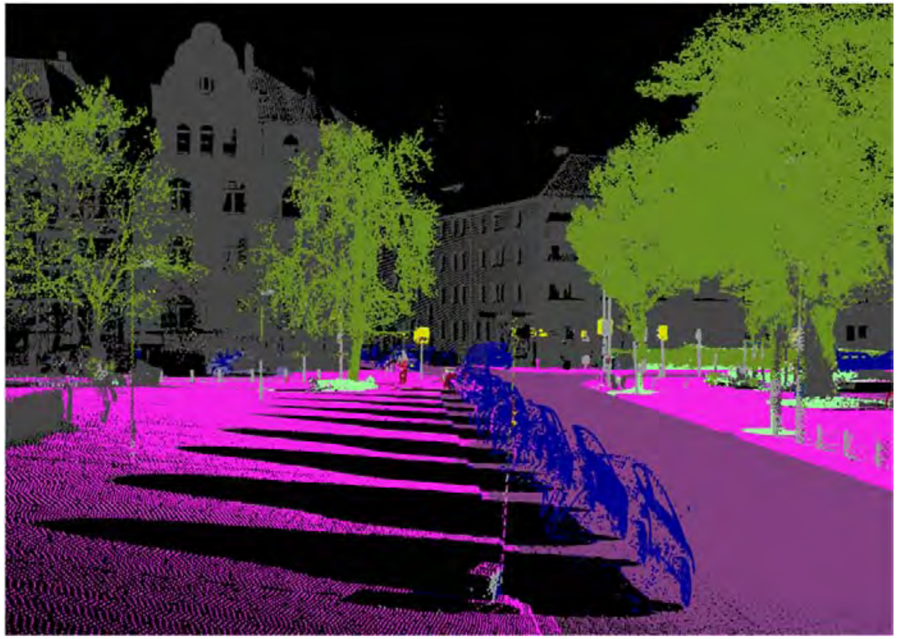
Unchanged dataset

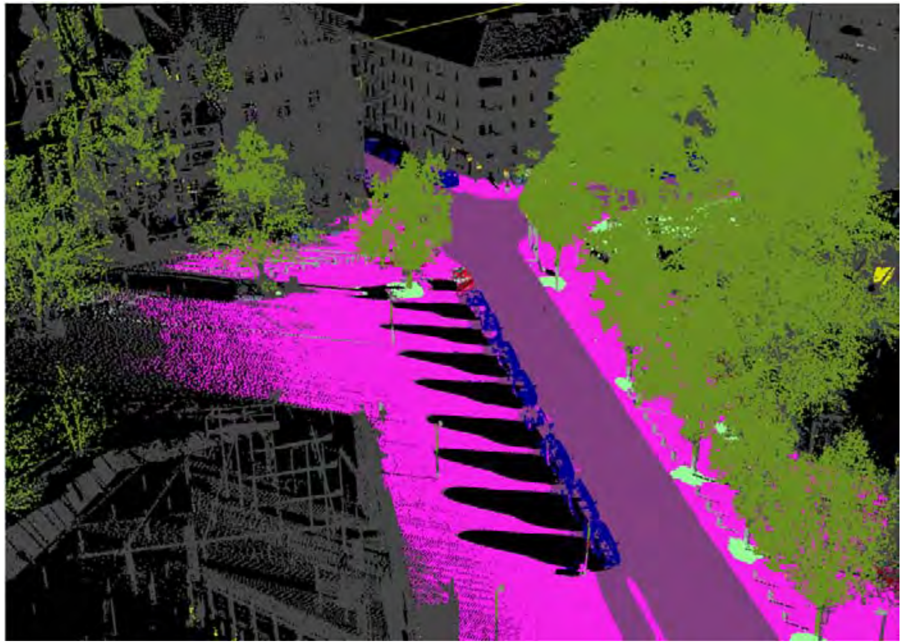


Dataset with flipped labels

Using transferred labels as training data

- We were able to successfully train a CNN on the 3D data
  - The CNN reached an estimated IoU of ~59.3%





i.c.sens

Khajeh Nasir Toosi

- [Warning: Wikipedia knowledge]
- Mathematician, architect, philosopher, physician, scientist, and theologian
- Poetry: An example from one of his poems:

“Anyone who knows, and knows that he knows,
makes the steed of intelligence leap over the vault of heaven.

Anyone who does not know but knows that he does not know,
can bring his lame little donkey to the destination nonetheless.

Anyone who does not know, and does not know that he does not know,
is stuck forever in double ignorance.”

Unknown engineer

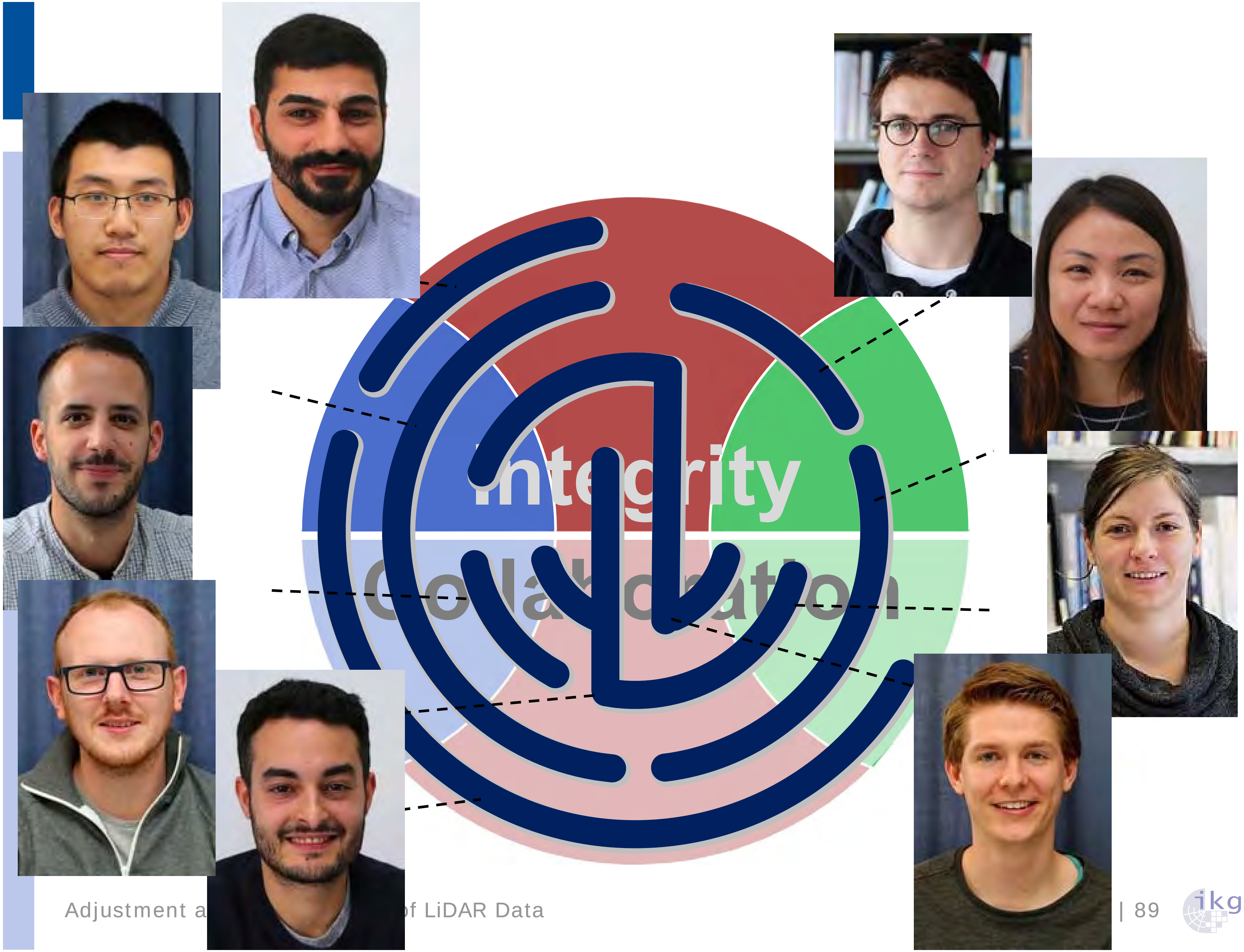
- If we are sure our system is working
  - We can go full speed ahead
- If we know it is not working
  - We may use a fail-safe mode to get to the goal nevertheless
- If our system is not working and we don't know
  - We're doomed!

- We need to know if our system is working within specifications
  - **integrity**
- Others may help us to determine this
  - **collaboration**

i.c.sens

Research Training Group

Integrity and Collaboration in Dynamic Sensor Networks


Integrity
Collaboration

„Mapathon“ (June, 12, 2017)

„Mapathon“ (June, 12, 2017)

Summary

- LiDAR alignment
  - Large numeric problem (HPC) → Big data problem (Hadoop et al.)
  - Relevant for crowd-based map acquisition
- Change detection
  - Robotics problem, relevant for lifelong mapping, maps for SDC
- Interpretation
  - Graph-based geometric approach, robust w.r.t. appearance
  - Transfer learning: images and LiDAR, using Cityscapes + deep learning to transfer labels
- DFG Research Training Group: i.c.sens
  - Integrity and collaboration in dynamic sensor networks
  - Alternative error measures, image and LiDAR interpretation
  - (Next PhD application round coming soon).