University of Stuttgart
Institute for Photogrammetry

Sparse 3D CNNs for citywide semantic mapping with ALS

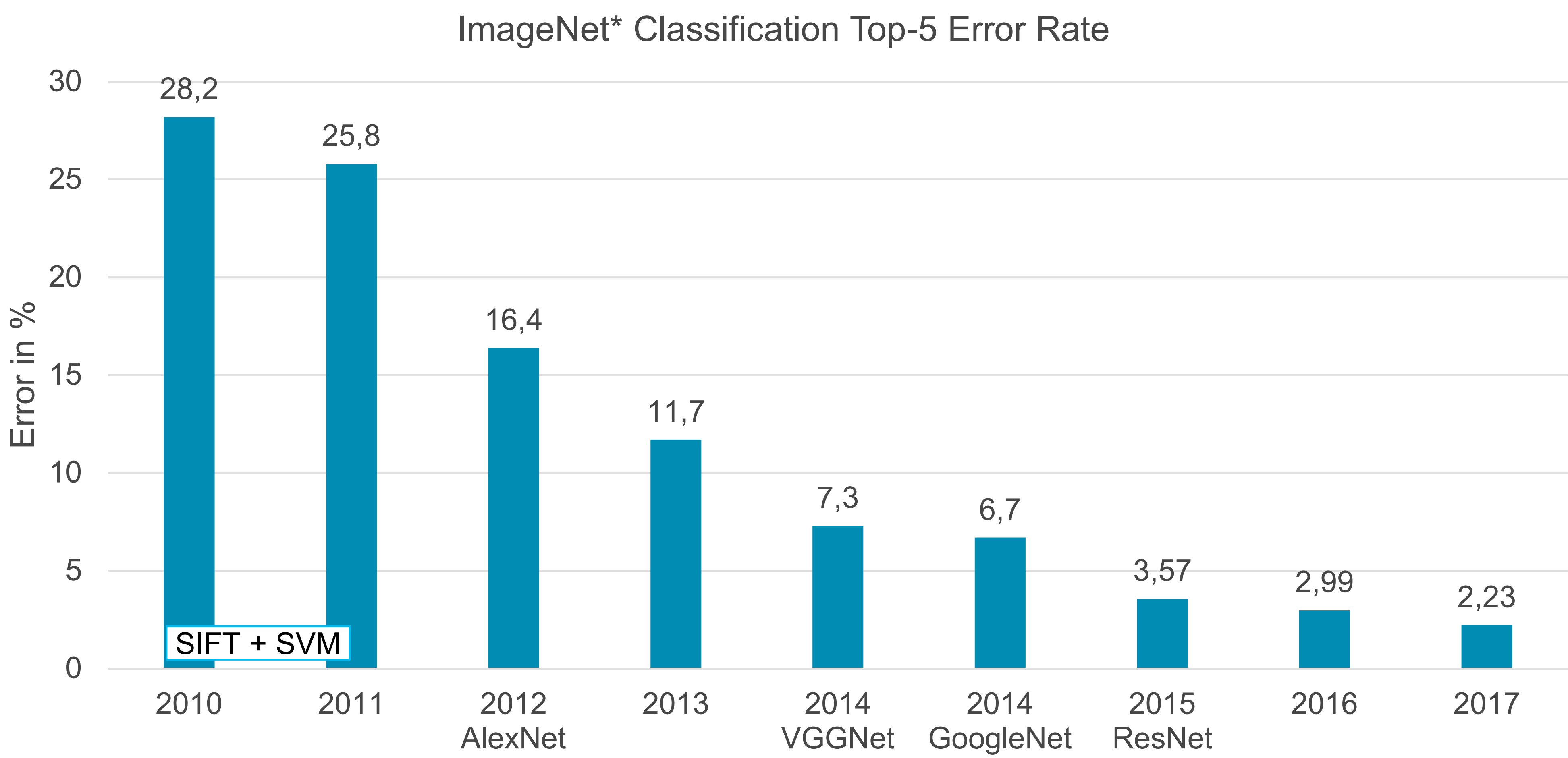
Stefan Schmohl

2nd International Workshop on *Point Cloud Processing*

University of Stuttgart, Dec. 4-5, 2019

EuroSDR DGPF ifp

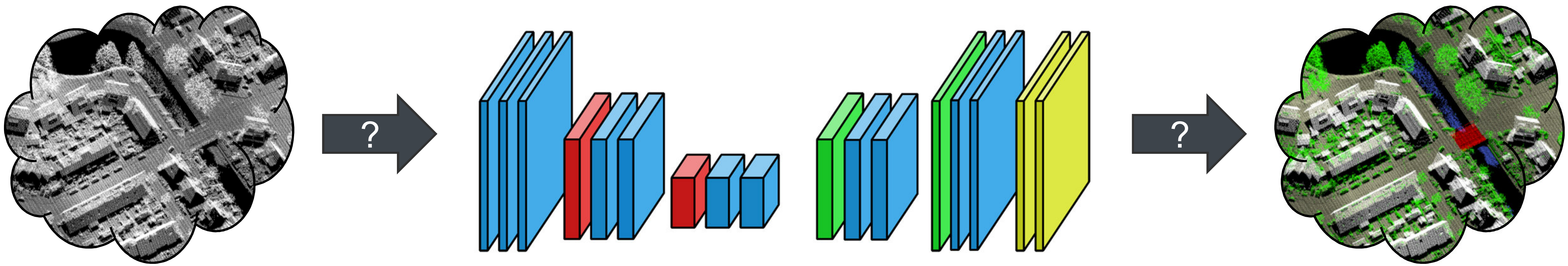
Motivation: CNNs pushing forward the progress in vision tasks



http://image-net.org

Russakovsky et al., 2015: *ImageNet: Large Scale Visual Recognition Challange*

CNNs – how to apply them to point clouds?



➔ **Challenges**:

point clouds are…

a) **three dimensional**

b) **irregular**

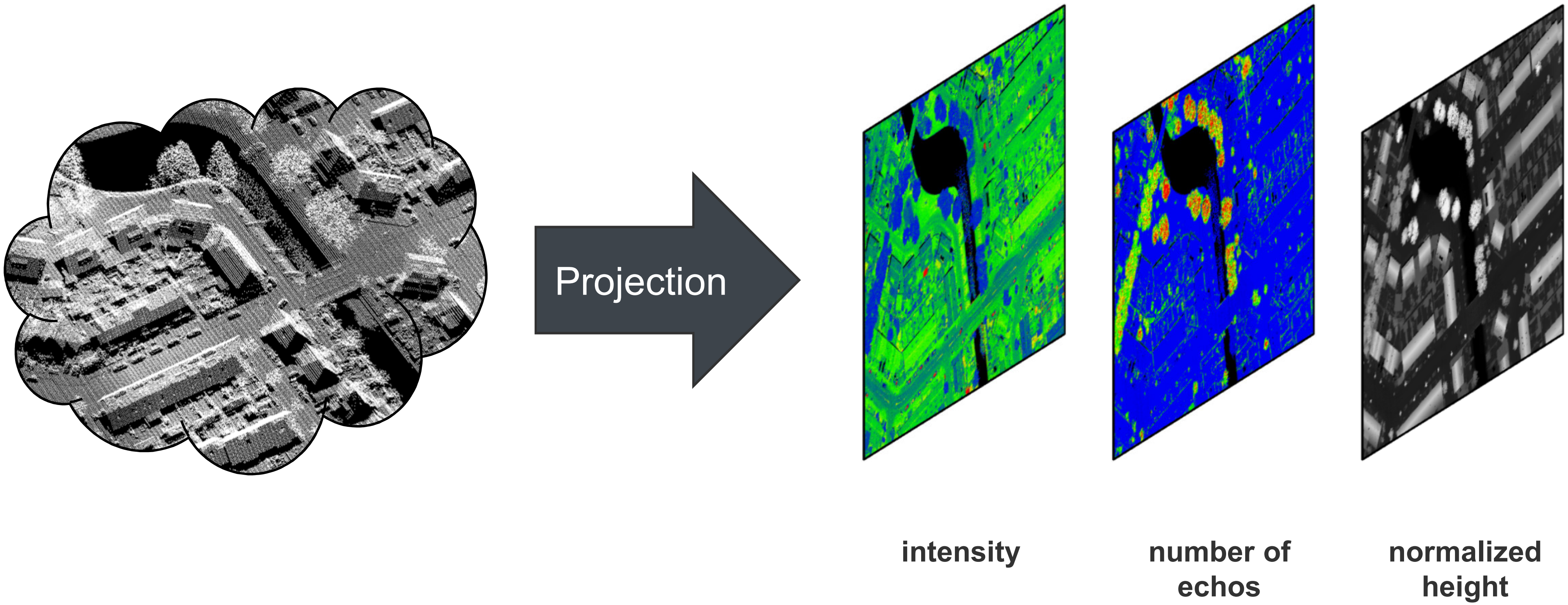
Convolutional Layer

Strided Convolution

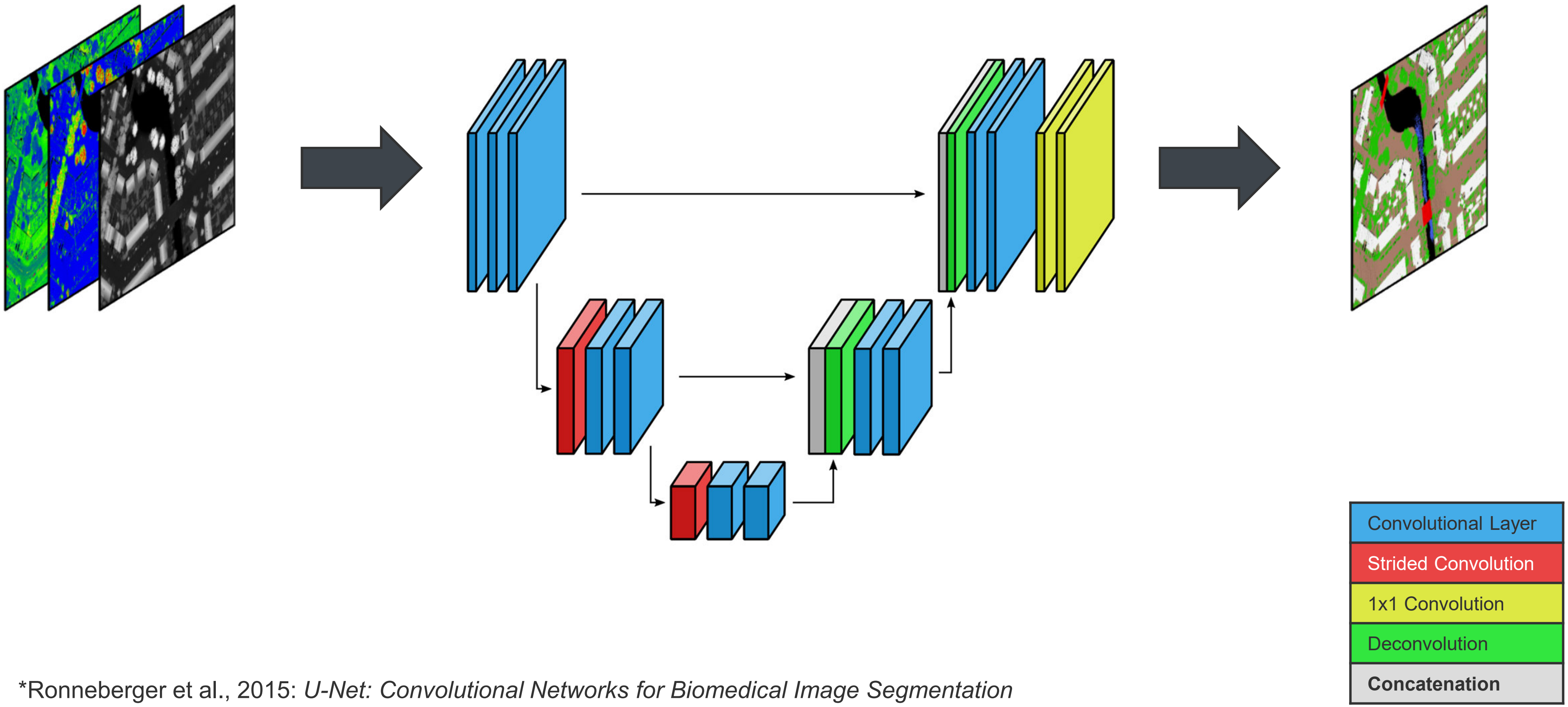
1x1 Convolution

Deconvolution

2D Rasterization

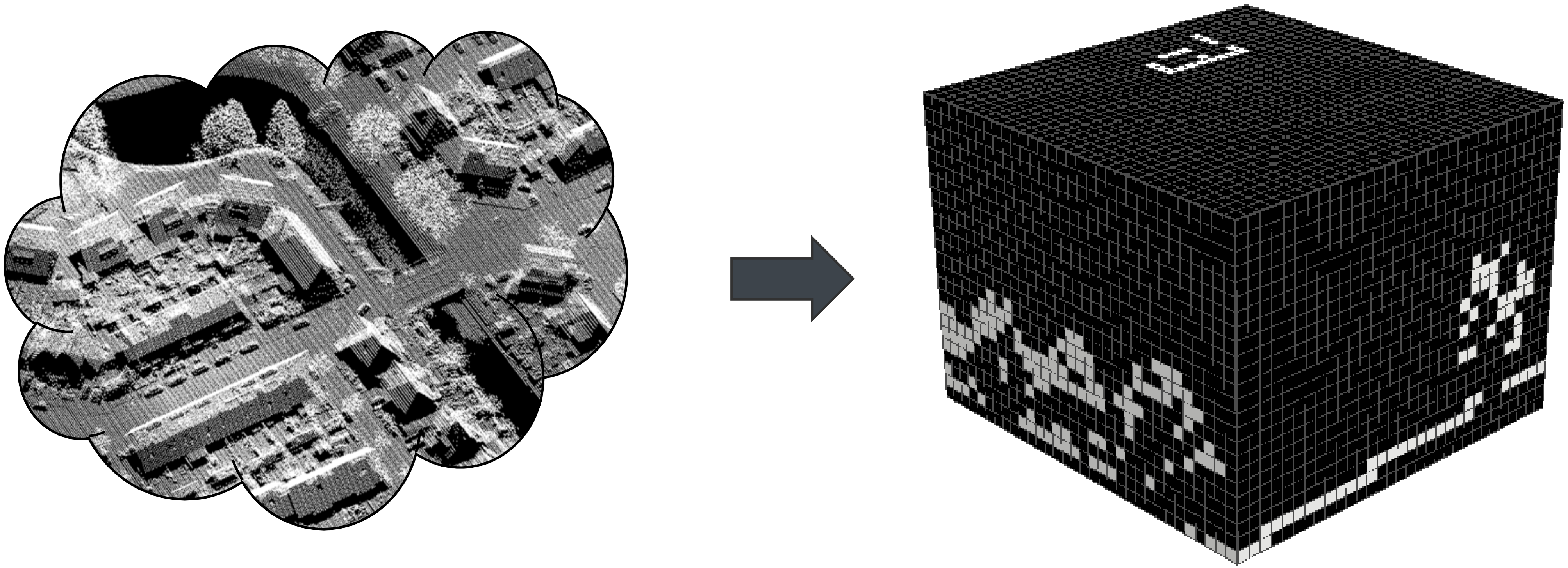


2D U-Net*

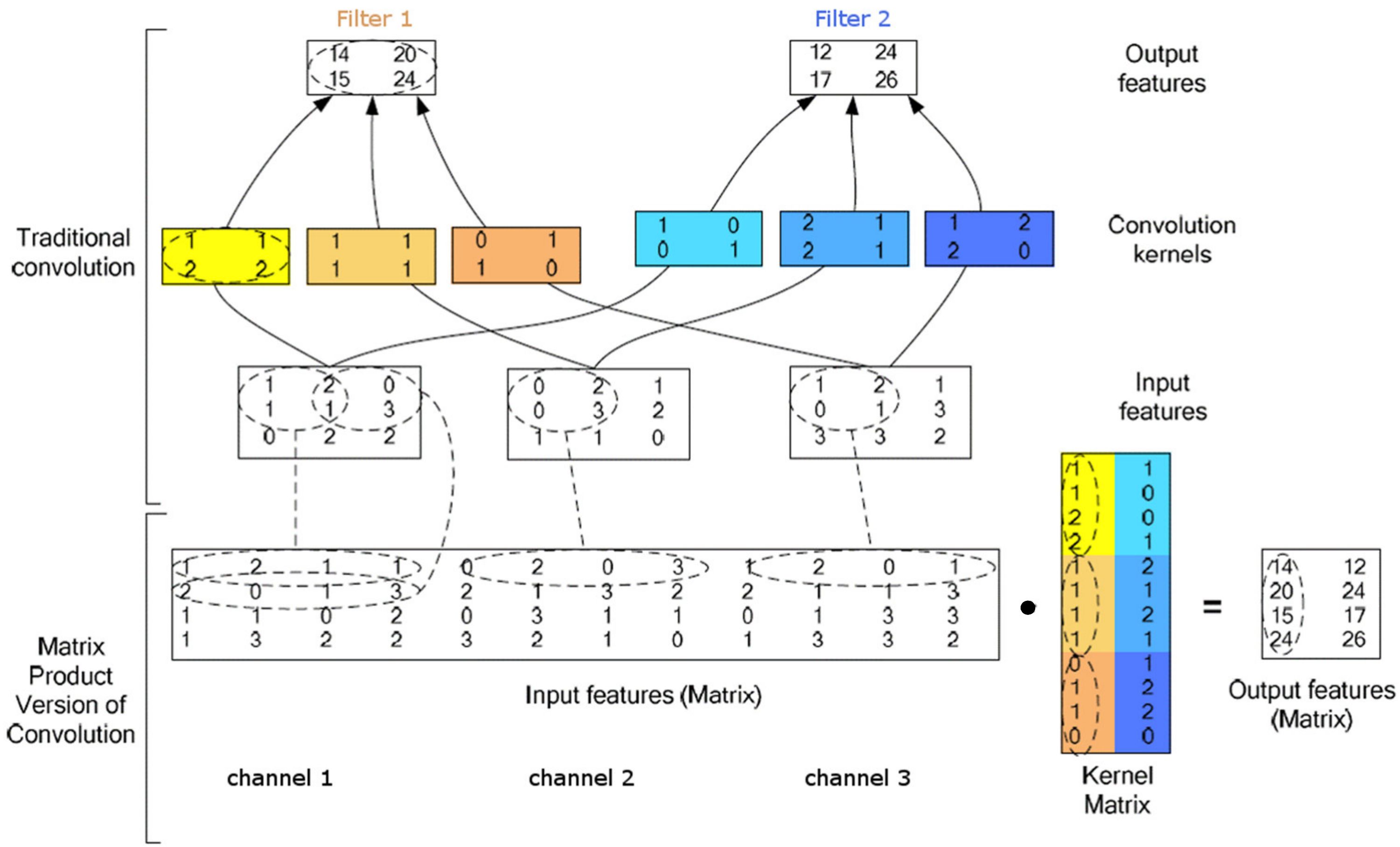


*Ronneberger et al., 2015: *U-Net: Convolutional Networks for Biomedical Image Segmentation*

3D CNNs for voxelized point clouds?



Convolution by GEneral Matrix Multiply (GEMM)



$$X \cdot W = Y$$

Chellapilla et al., 2006: *High Performance Convolutional Neural Networks for Document Processing*

Convolution by GEneral Matrix Multiply (GEMM)

$$Y = X * W$$

$$\Downarrow$$

One row per spatial kernel location → $\boldsymbol{Y} = \boldsymbol{X} \cdot \boldsymbol{W}$

↑

One column per filter = output channel

Chellapilla et al., 2006: *High Performance Convolutional Neural Networks for Document Processing*

Sparse 3D CNN

Idea 1: input data structure does not matter

=> construct $\boldsymbol{X}$ out of list of coordinates (i.e. "voxel cloud")

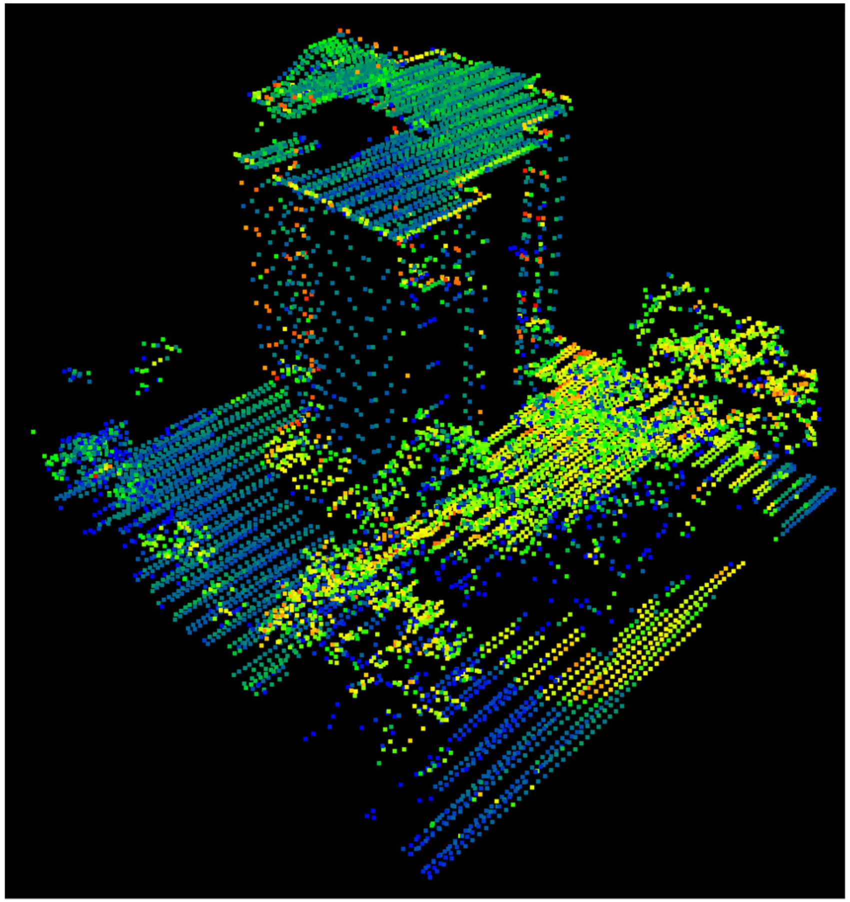
Idea 2: sparse GEMM

=> construct $\boldsymbol{X}$ without the "empty" locations (or rows, respectively).

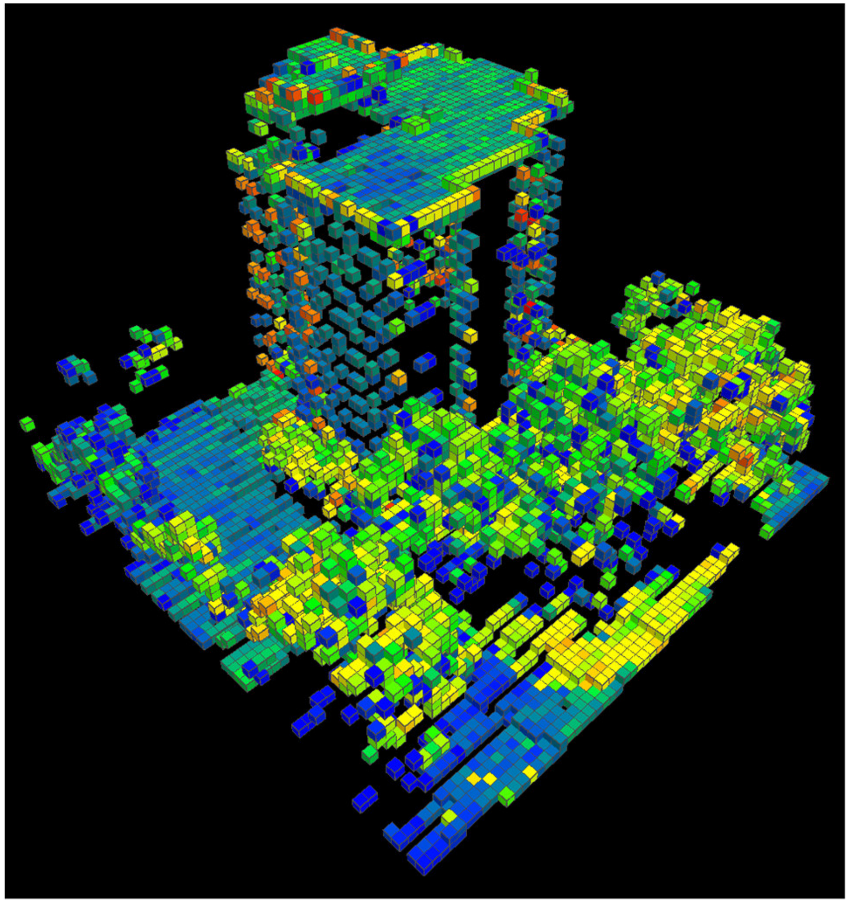
=> Submanifold Sparse Convolutional Networks*

*Graham et al., 2018: *Semantic Segmentation with Submanifold Sparse Convolutional Networks*

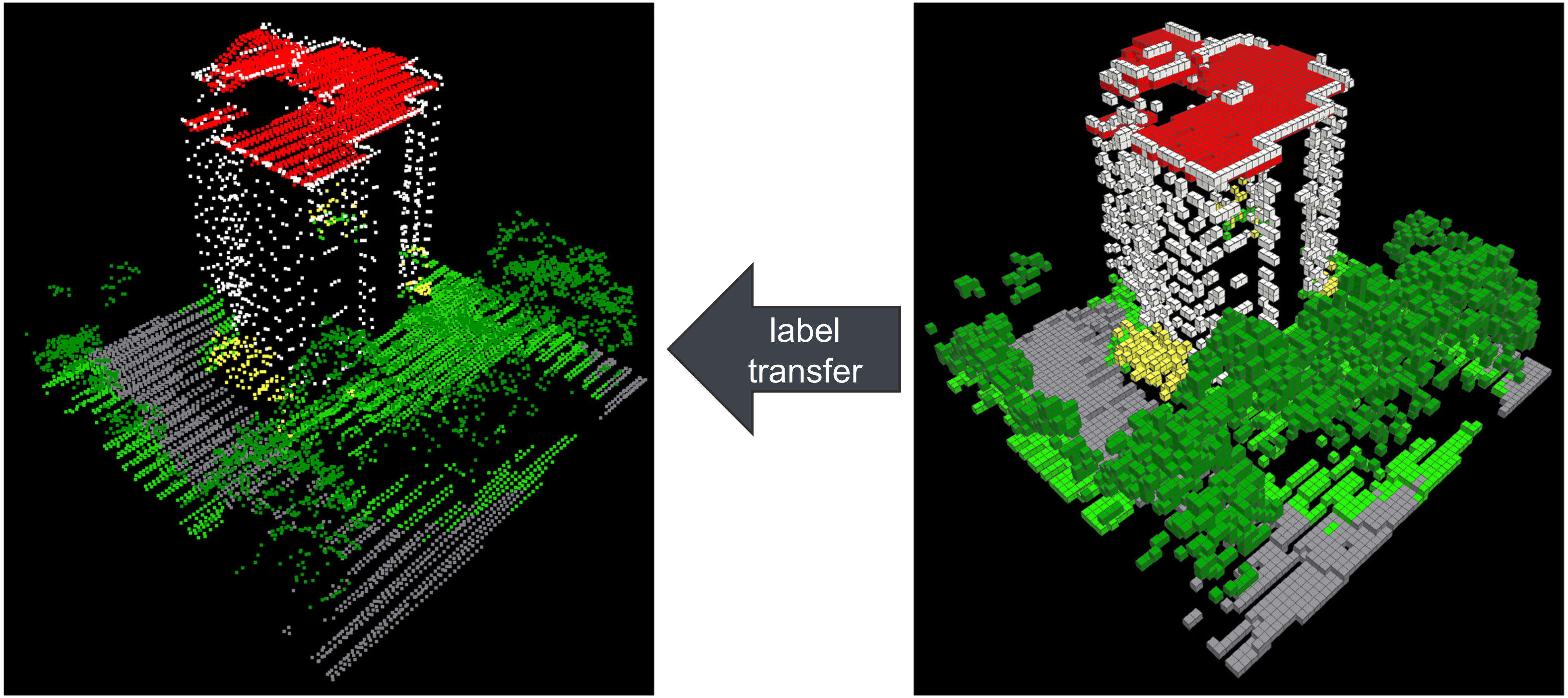
Voxelization



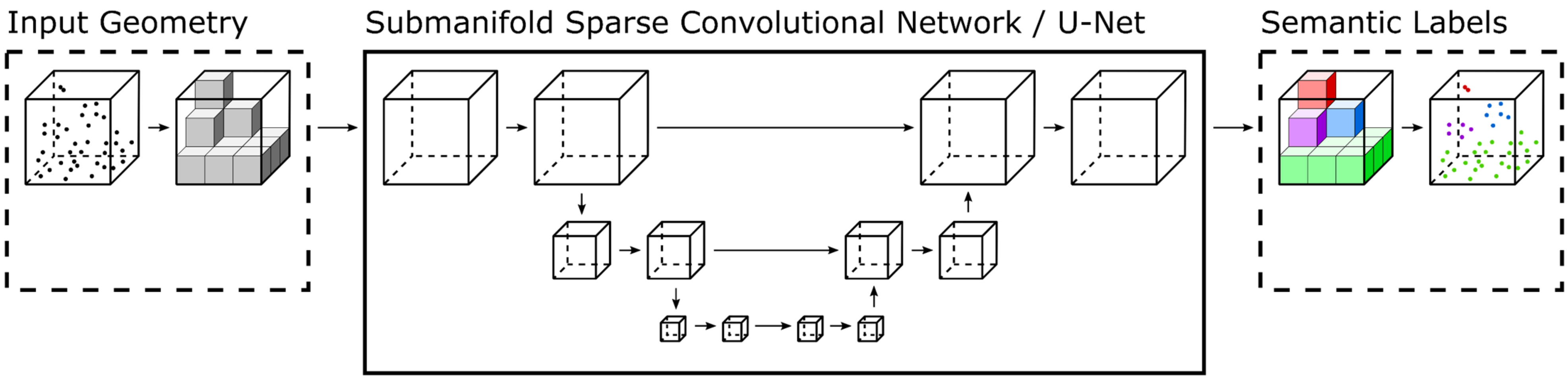
averaging
0,5 m



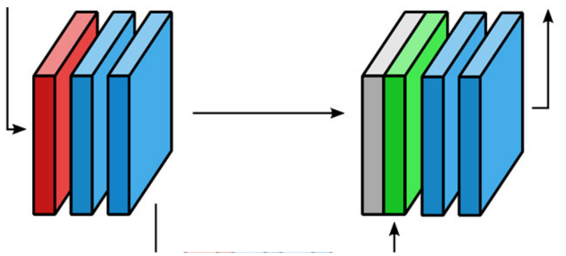
Voxelization



Network Architecture



6-7 U-Net levels:

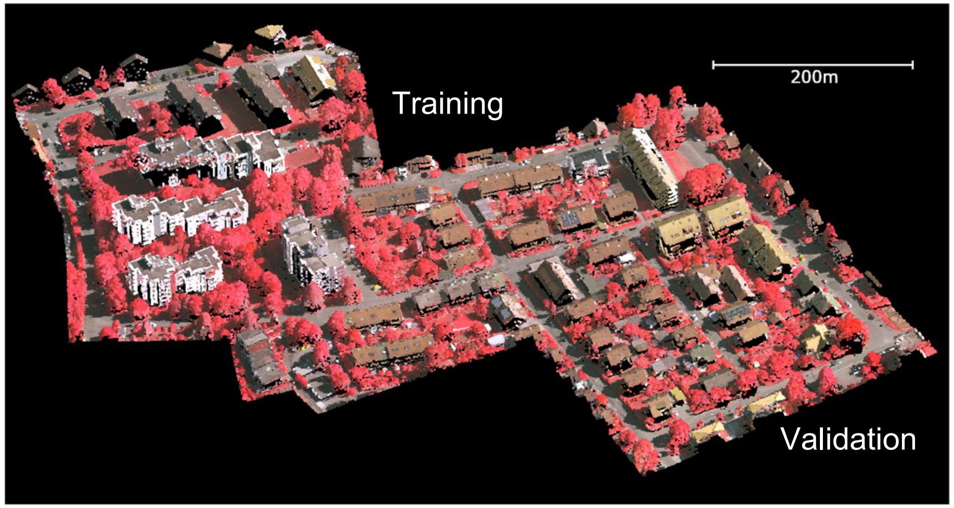


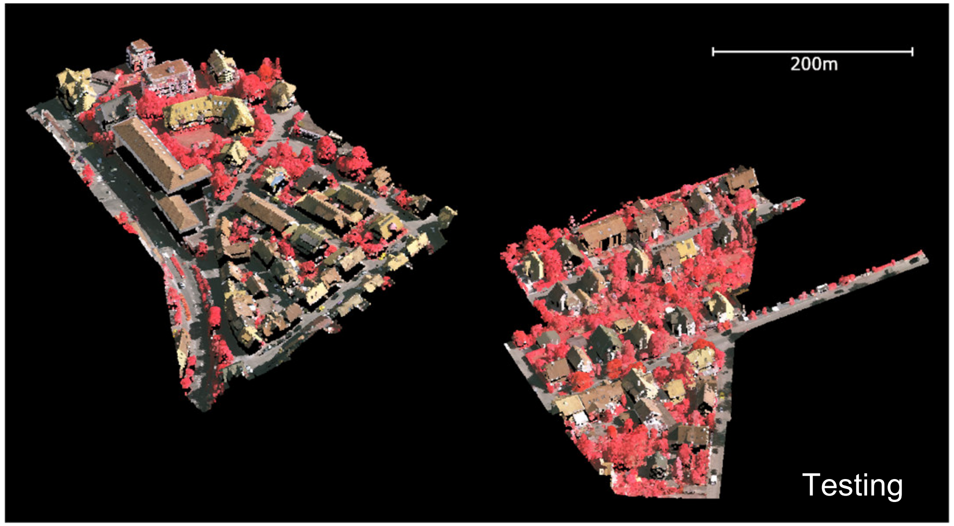
Convolutional Layer	Including: • Batchnorm • ReLU
Strided Convolution	
Deconvolution	
Concatenation	

Dataset I - ISPRS 3D Semantic Labeling (*Vaihingen 3D*)

- Training area: ~ 700.000 points
- Testing area: ~ 400.000 points
- Point density: ~ 4-8 points / m²
- Features:
  - intensity
  - echo number
  - number of echos
  - CIR
- 10 semantic classes

- Voxel size 0.5 m





Results on V3D

<u>Features</u>:

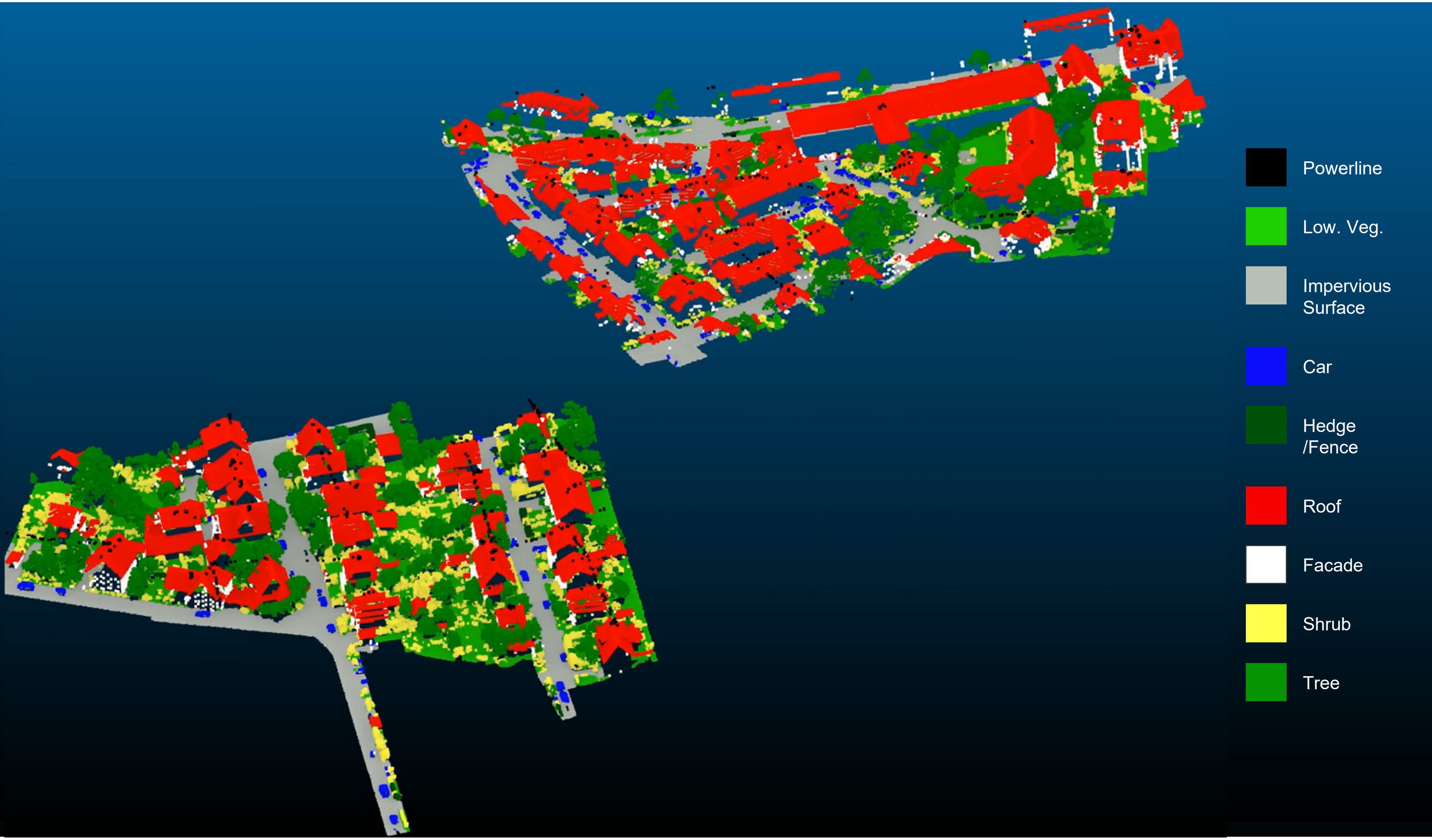
- intensity
- echo number
- # of echos
- CIR

<u>Voxel size</u>:

0.5 m

⇒ OA = **85.0 %**

⇒ Time: **11 s**



Memory (and Runtime) on V3D

Graphics memory	Voxel size				
	2.0 m	1.0 m	**0.5 m**	0.25 m	0.125 m
dense	1.5 GB	7.7 GB	**-**	-	-
sparse	0.9 GB	1.5 GB	**2.2 GB**	4.9 GB	7.9 GB

Dataset II - Dutch National Height Model (*AHN3*)

- Training area: ~ 20.000.000 points
- Testing area: ~ 40.000.000 points
- Point density: ~ 9-16 points / m²
- Features:
  - intensity
  - echo number
  - number of echos
  - scan angle
- 5 semantic classes

- Voxel size 0.25 m

Results on AHN3

<u>Features</u>:

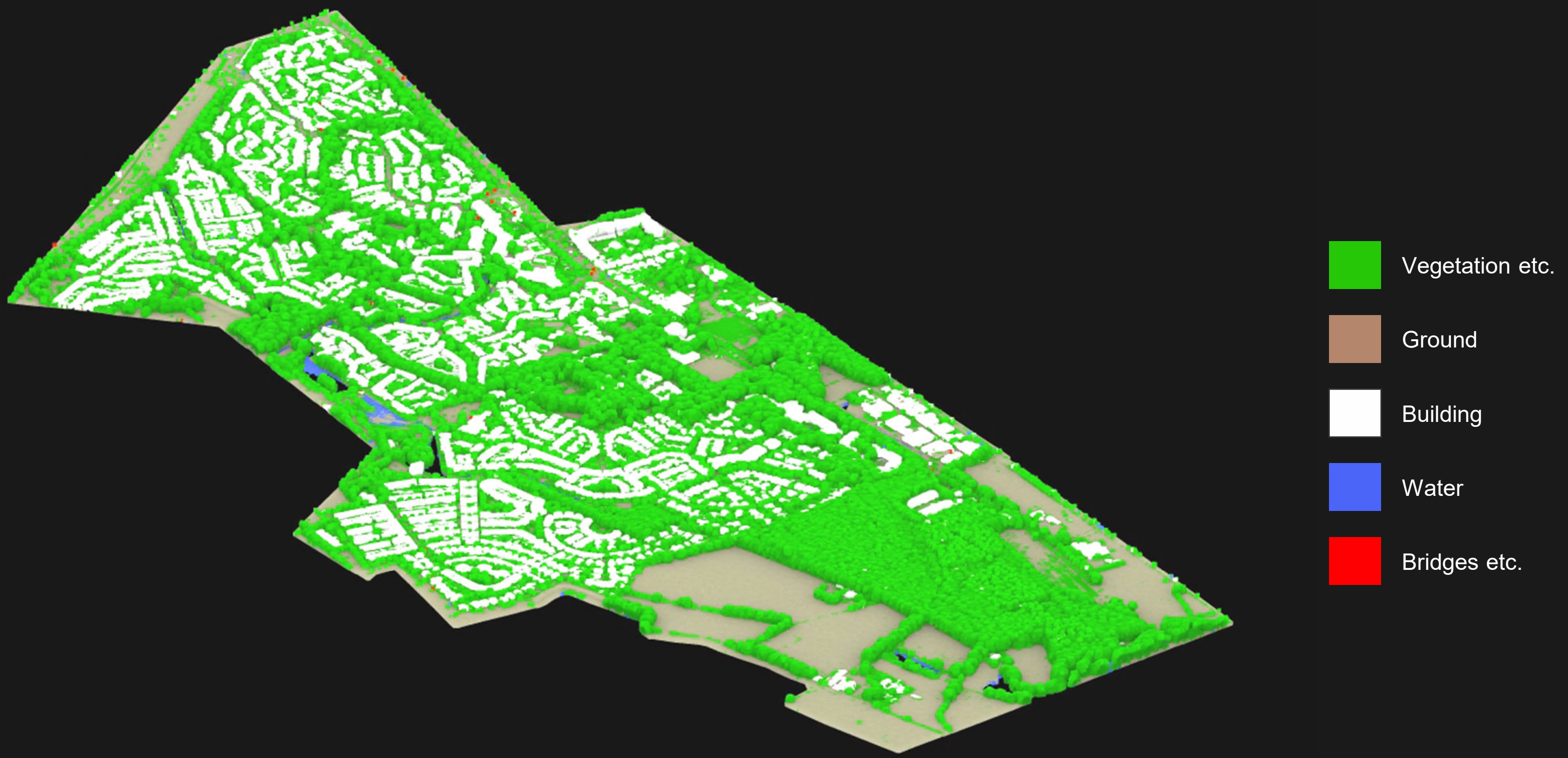
- intensity
- echo number
- # of echos
- scan angle

<u>Voxel size</u>:

0.25 m

⇒ OA = **96 %**

⇒ Time: **108 s**



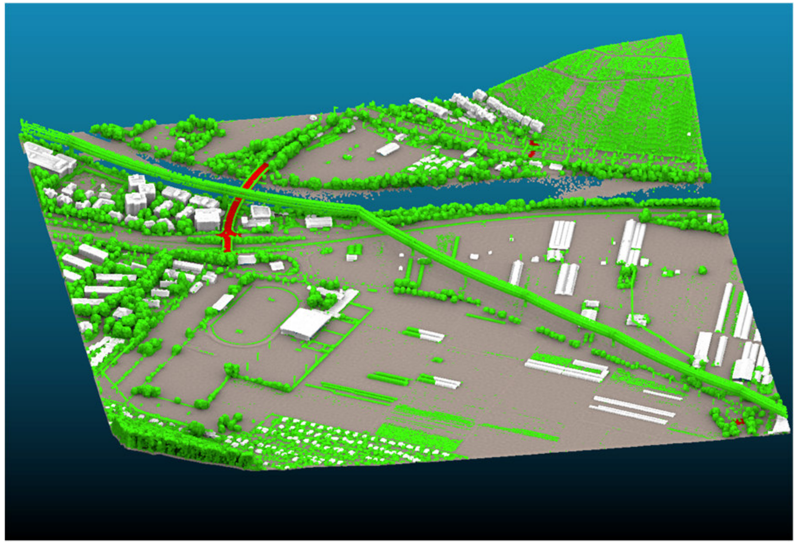
Dataset III - Stuttgart

Landesamt für
Geoinformation und Landentwicklung

Baden-Württemberg

Let's go even bigger!

- Point cloud properties & ground truth similar to AHN3.
  - however not the data domain, obviously.
- Coverage of Stuttgart:
  - 337 tiles, each 1 km²
- Goal:
  - Semantic classification of Stuttgart (with more than ground truth classes)
- Idea:
  - Find appropriate training data.



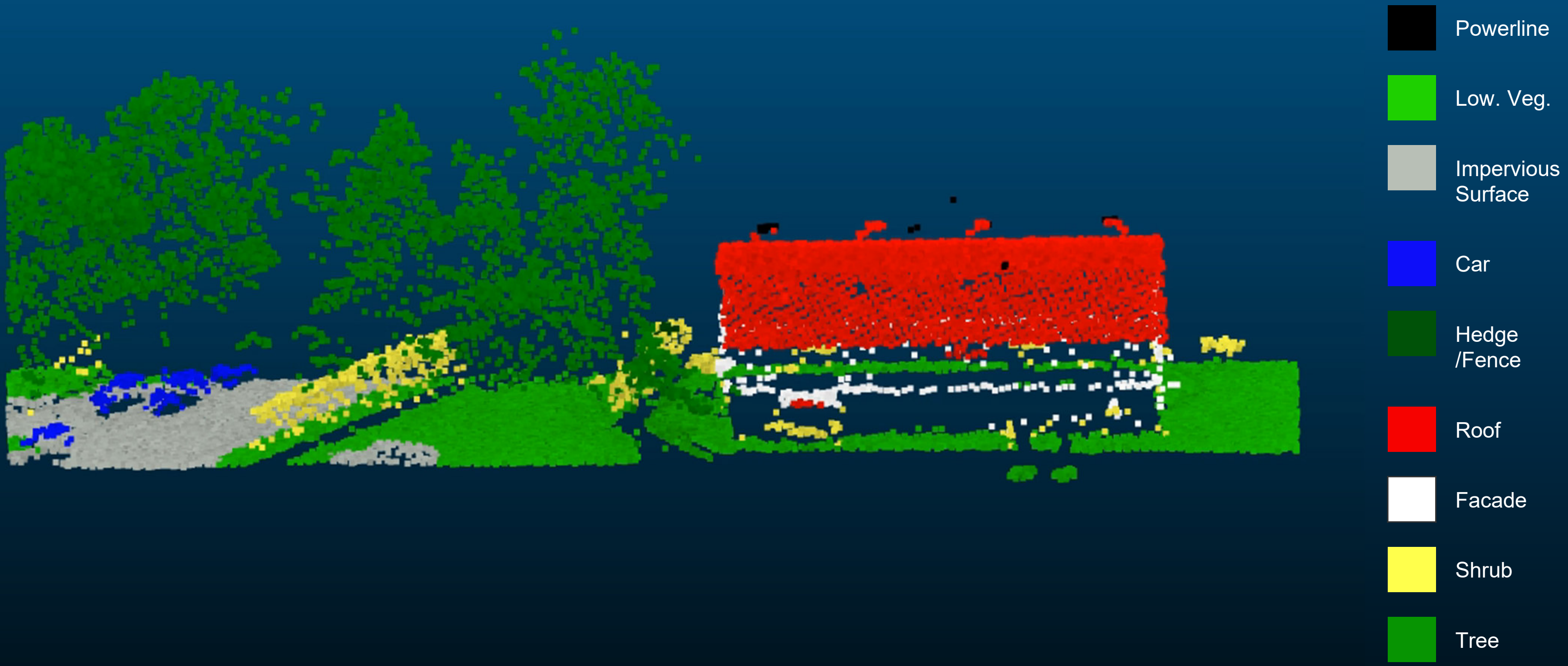
V3D → Stuttgart

<u>Features</u>:

- intensity
- echo number
- # of echos

<u>Voxel size</u>:

0.5 m



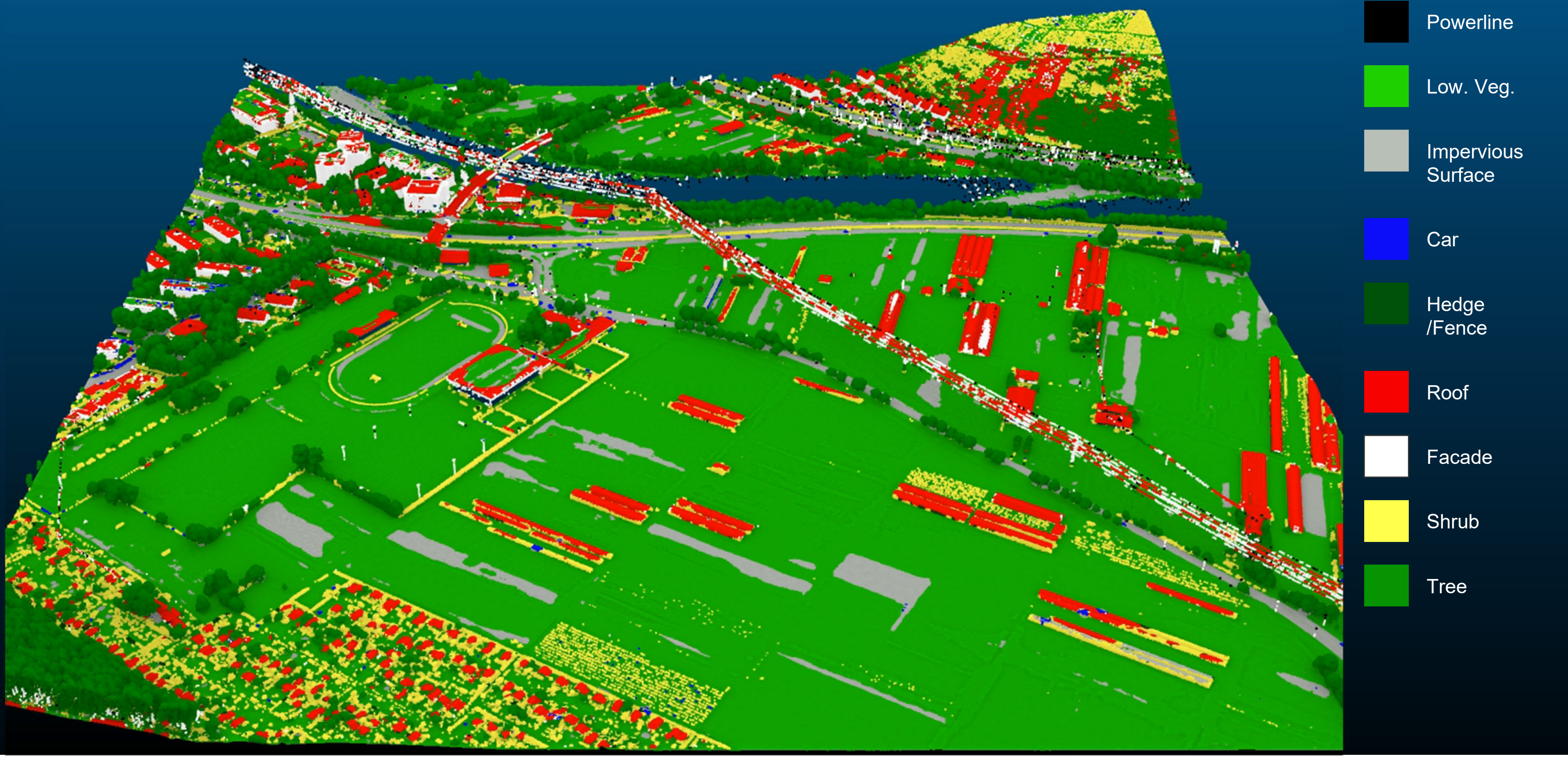
V3D → Stuttgart

<u>Features</u>:

- intensity
- echo number
- # of echos

<u>Voxel size</u>:

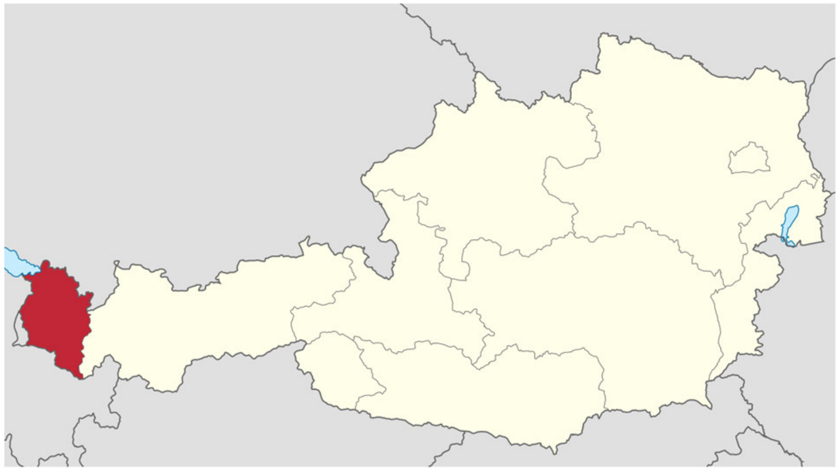
0.5 m



Dataset IV – Vorarlberg, Austria

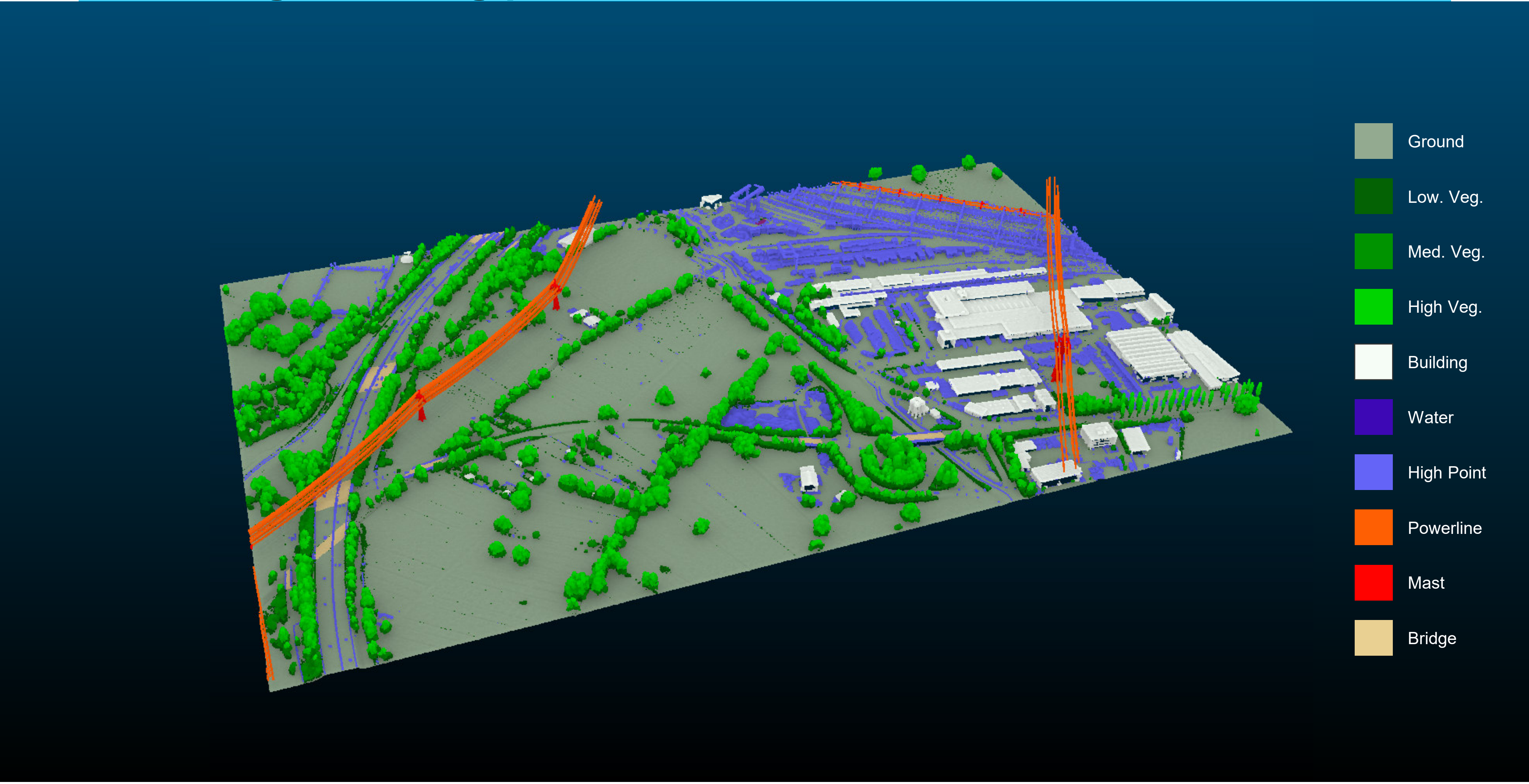
Landesvermessungsamt Vorarlberg

- Point cloud properties & **data domain** similar to Stuttgart.
  - But higher class diversity!
- Coverage over entire Vorarlberg:
  - 536 tiles, each 6,25 km²
- Selected training area:
  - Parts of the city of Bregenz & vicinity
  - ~ 63 M points
  - ~ 3 km²

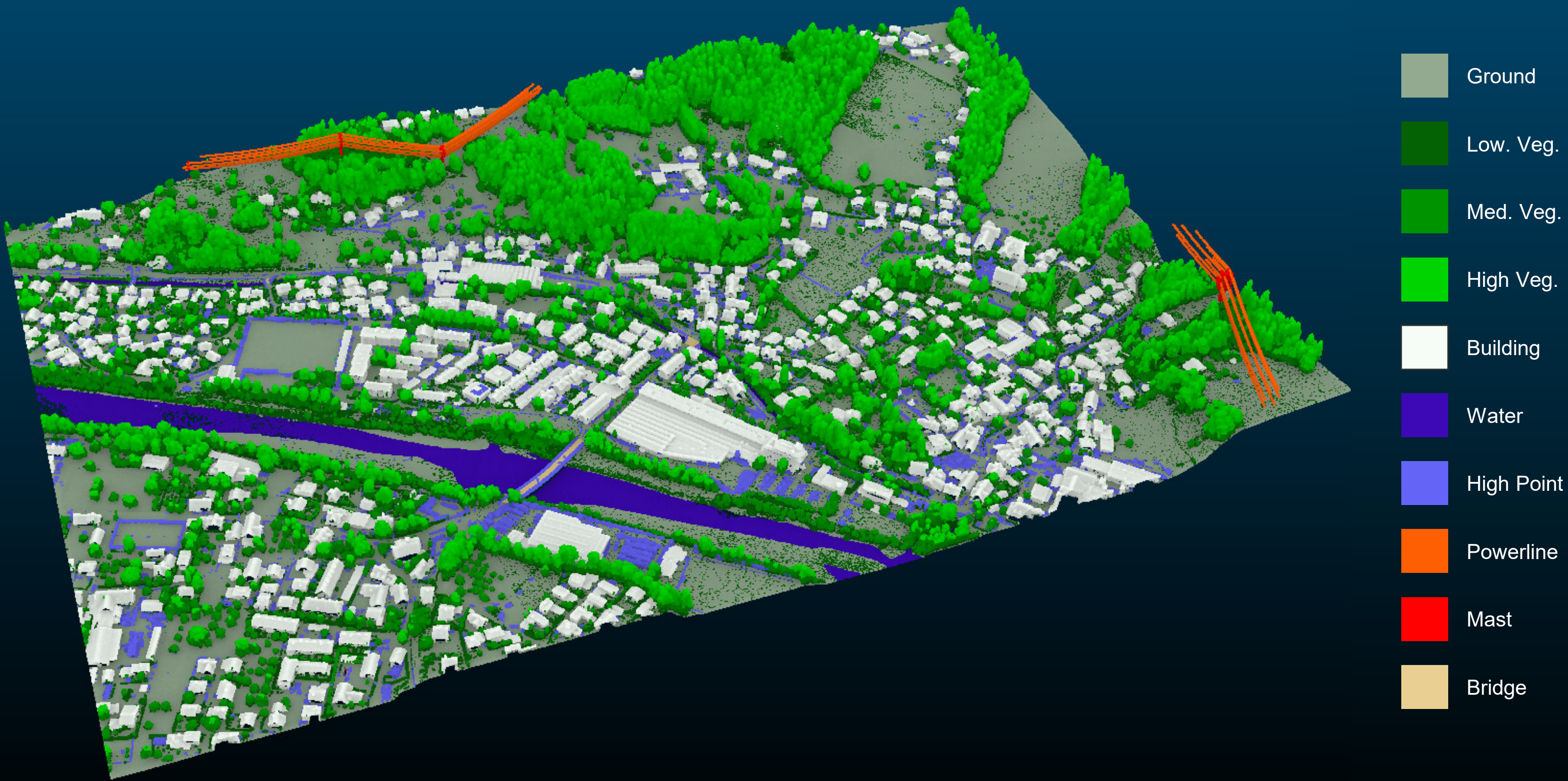


https://de.wikipedia.org/wiki/Vorarlberg

Vorarlberg - Training part 1

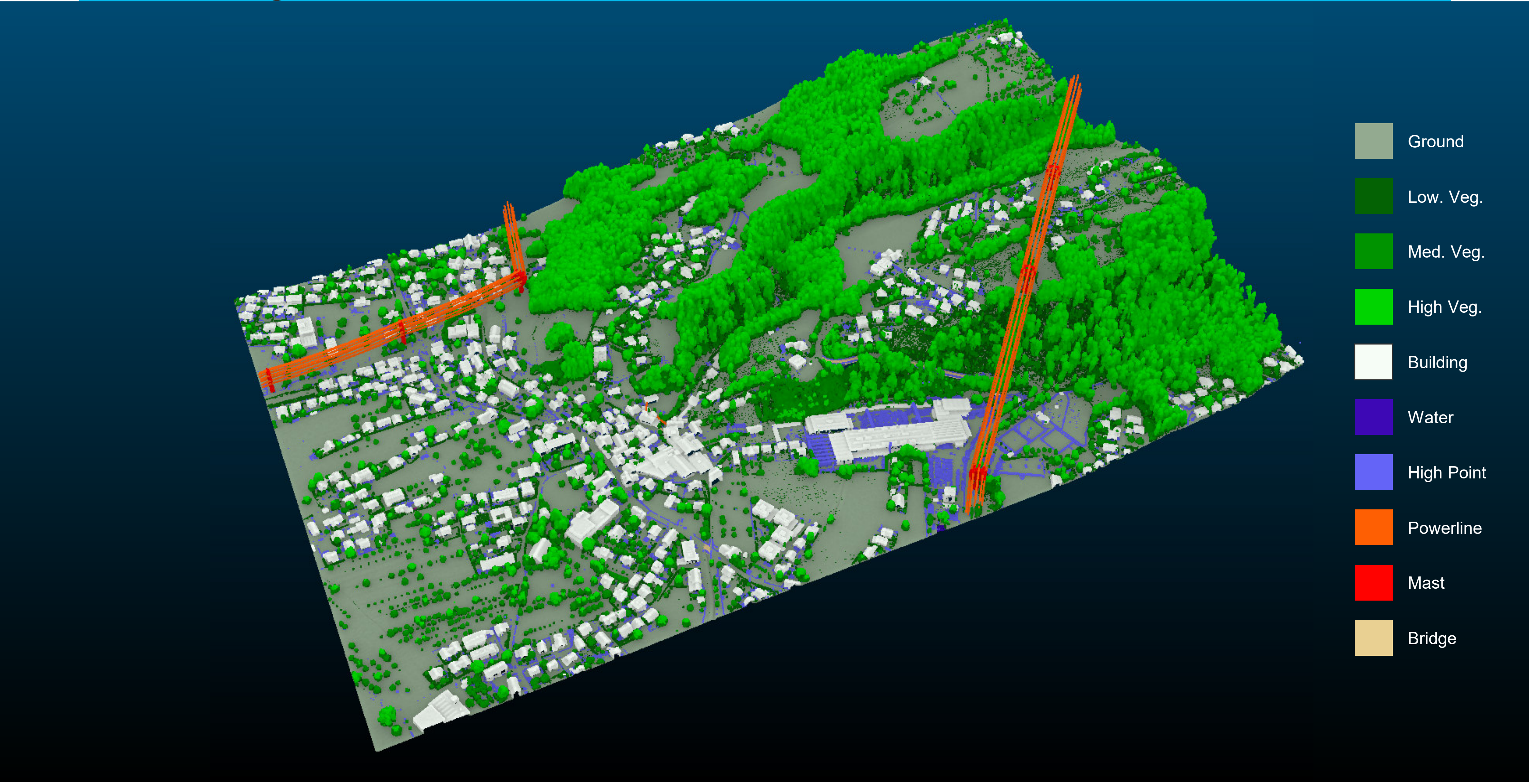


Vorarlberg - Training part 2



Vorarlberg - Training part 3

Vorarlberg - Validation



City wide mapping? Better buy a new graphics card!

12 GB, 3840 cores

24 GB, 4608 + 576 cores

=> updating drivers, cuda, PyTorch & sparse CNN framework

=> 3x slower!!

WWW.PHDCOMICS.COM

Vorarlberg → Stuttgart: First results

Features:

- intensity
- echo number
- # of echos
- scan angle

Voxel size:

0.25 m

27 M points

⇒ Time: **173 s**

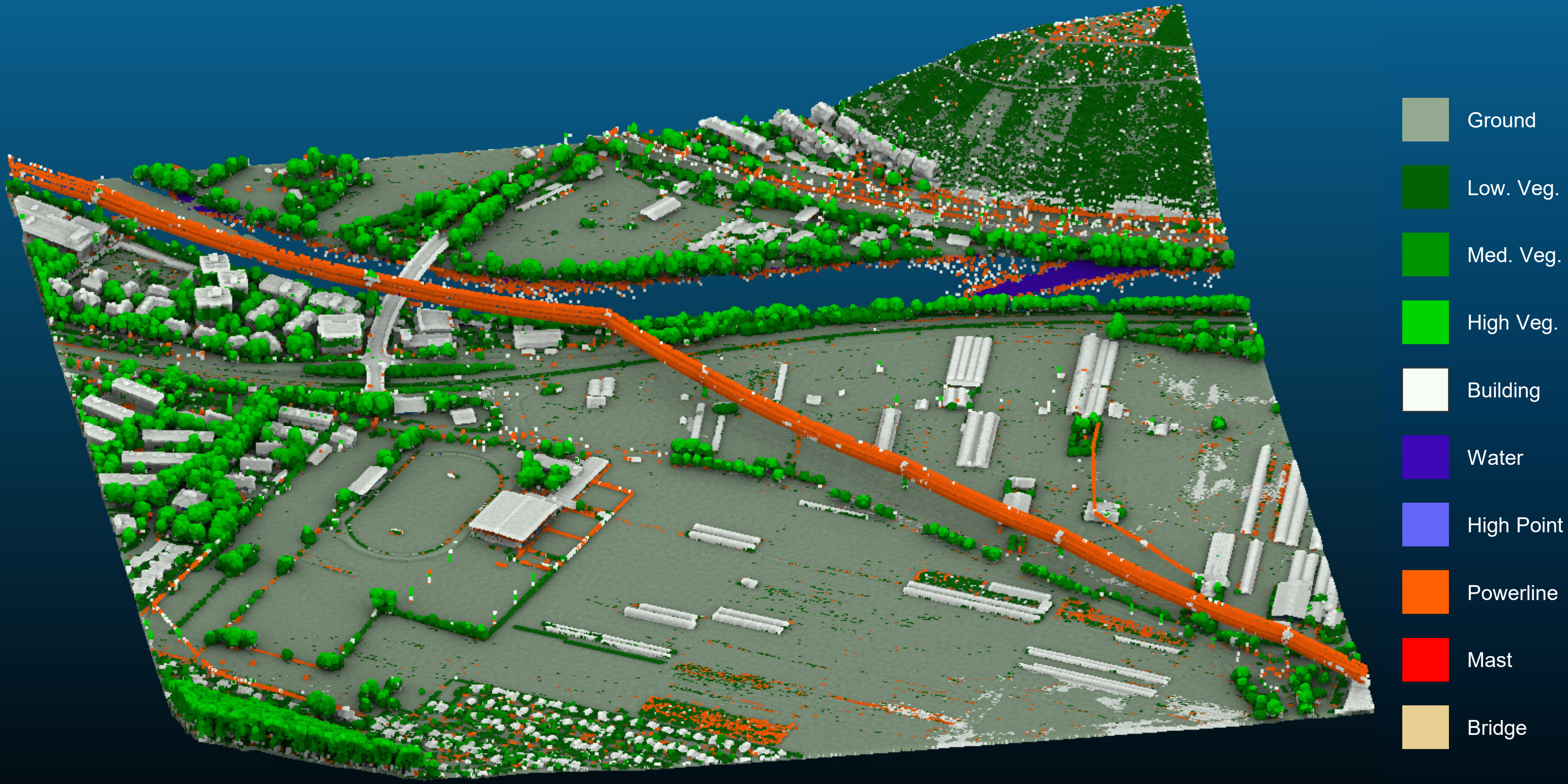
Vorarlberg → Stuttgart: First results

<u>Features</u>:

- intensity
- echo number
- # of echos
- scan angle

<u>Voxel size</u>:

0.25 m



Vorarlberg → Stuttgart: First results

<u>Features</u>:

- intensity
- echo number
- # of echos
- scan angle

<u>Voxel size</u>:

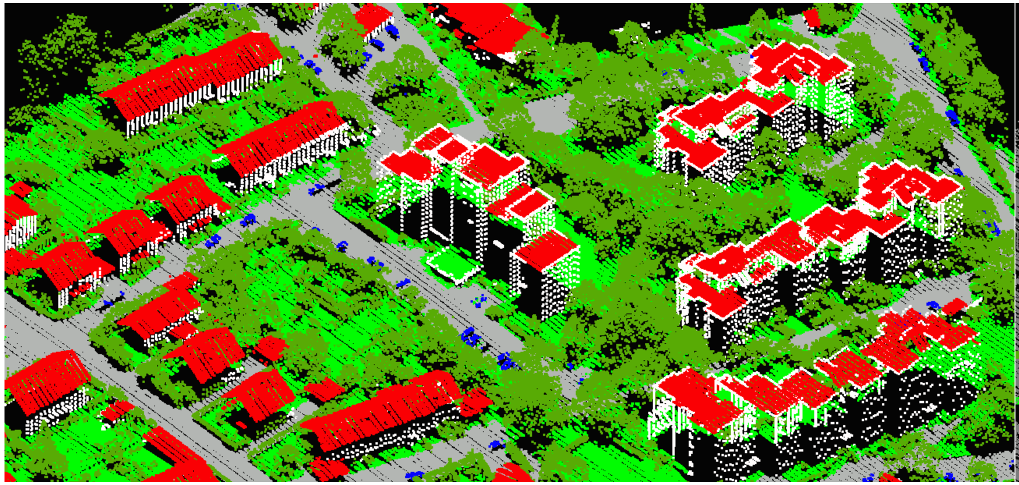
0.25 m

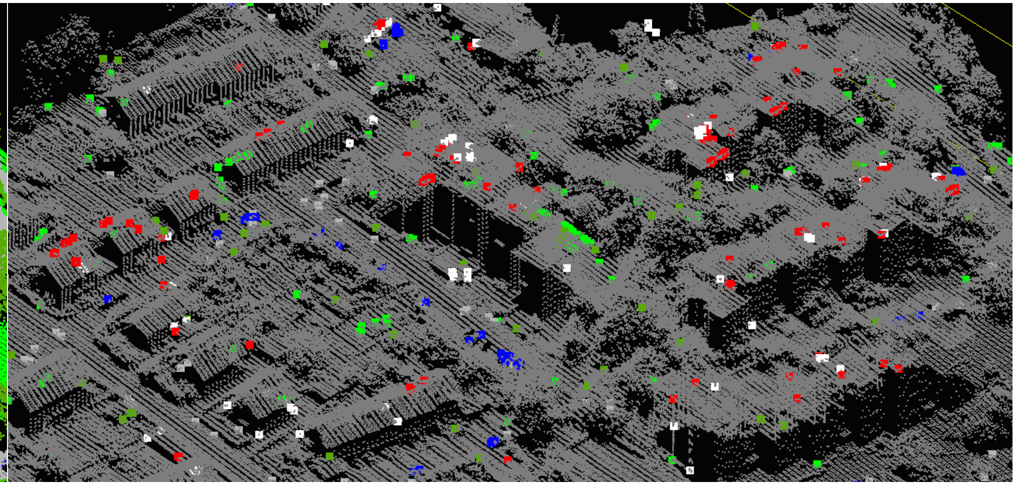
Ground
Low. Veg.
Med. Veg.
High Veg.
Building
Water
High Point
Powerline
Mast
Bridge

Dense vs. *sparse* training dataset

Now with a sparse 3D CNN!

dense training set from expert | VS. | *sparse* training set from crowd





- $100\ \%$ of points labeled
- $\mathrm{OA} = \mathbf{88.39}\ \%$
- Costs = ???
- Expert(s) actively involved in labeling

- $0.4\ \%$ of points labeled
- $\mathrm{OA} = \mathbf{85.43}\ \%$
- Costs < $130\ \$$
- Only providing data and wages

Conclusion

- No need for extra feature extraction or point-to-image conversion.
- Implicit geometry is the most important feature.
- State-of-the-art accuracy on V3D.
- Very fast inference.
- Good results also on sparsely labeled training data.

- Not quite ready for production
- “smart” training strategy needed for huge training data
- Some “problematic classes”

Thank you for listening!

Many Thanks to:

- **International Society for Photogrammetry and Remote Sensing (ISPRS)**
- **Publieke Dienstverlening Op de Kaart (PDOK)**
- **Landesamt für Geoinformation und Landentwicklung Baden-Württemberg (LGL)**
- **Landesamt für Vermessung und Geoinformation Vorarlberg**

for proving the data that made this work possible!

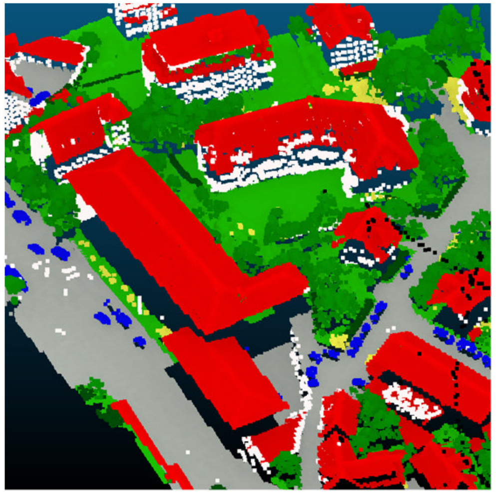
Universität Stuttgart

Supplementary

ifp

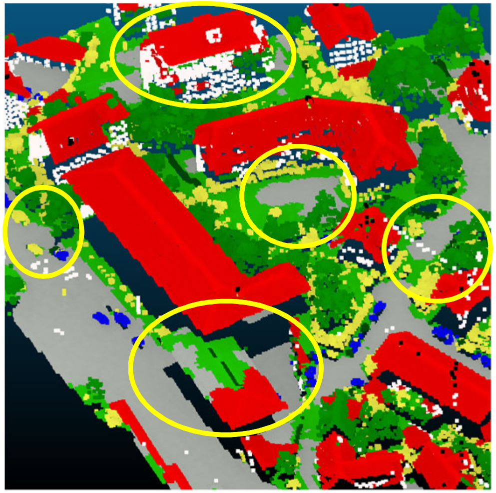
Results on V3D

ground truth



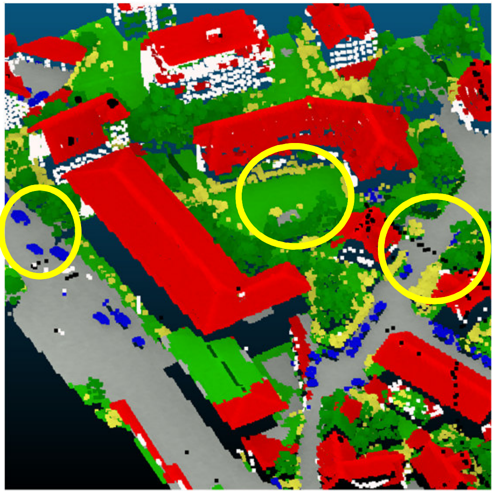
geometrie only
(voxel value = 1)

79,8%



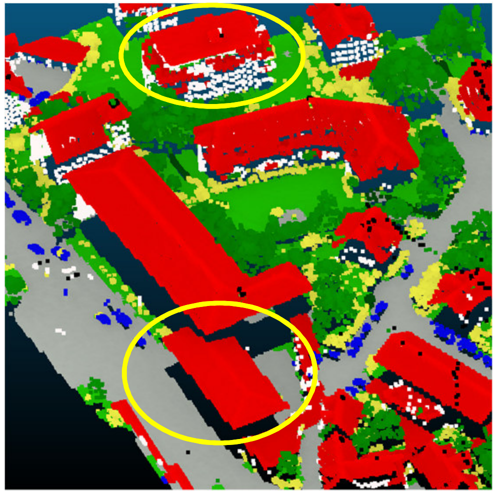
intensity,
echo number,
number of echos

84,2%



+ CIR

85,0%



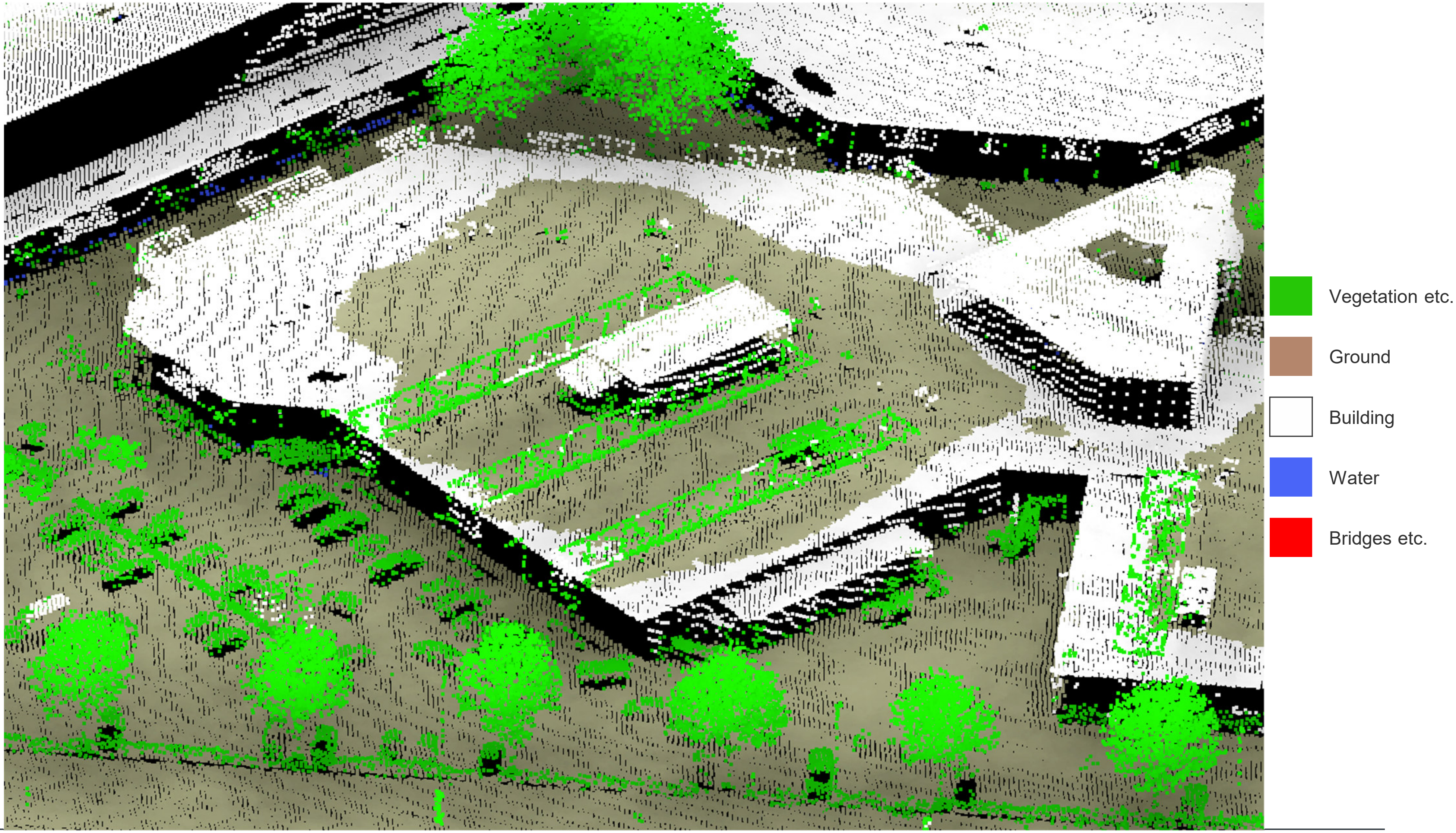
Powerline | Low. Veg. | Impervious Survace | Car | Hedge / Fence | Roof | Facade | Shrub | Tree

Runtime and Memory on V3D

	Voxel size				
	2.0 m	1.0 m	**0.5 m**	0.25 m	0.125 m
Graphics memory during training* (dense)	1.5 GB	7.7 GB	**-**	-	-
Graphics memory during training*	0.9 GB	1.5 GB	**2.2 GB**	4.9 GB	7.9 GB
Training time*	6 min	14 min	**30min**	63 min	107 min
Inference time*	0,3 s	0,4 s	**0,8 s**	1,4 s	2,0 s
Accuracy**	80,3 %	83,7 %	**84,2 %**	84,2 %	83,2 %

* per network. For each configuration, results of 10 independent networks were used as an ensemble.
** without CIR

Results on AHN3 – typical misclassifications



Features:

- intensity
- echo number
- # of echos
- Scan-Angle

Voxel size:

0,25 m

⇒ OA = **96%**

Results on AHN3 – typical misclassifications

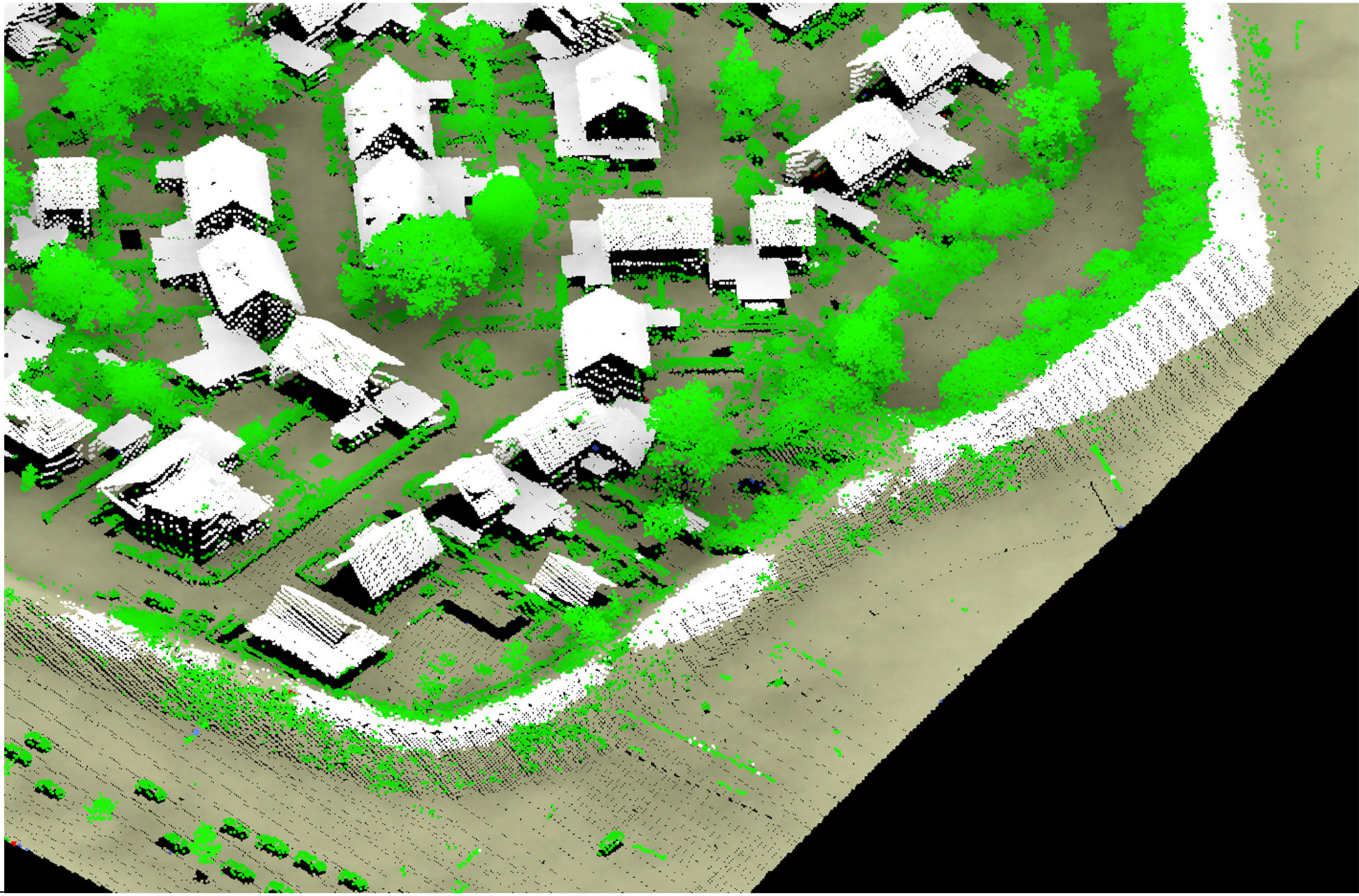
<u>Features</u>:

- intensity
- echo number
- # of echos
- Scan-Angle

<u>Voxel size:</u>

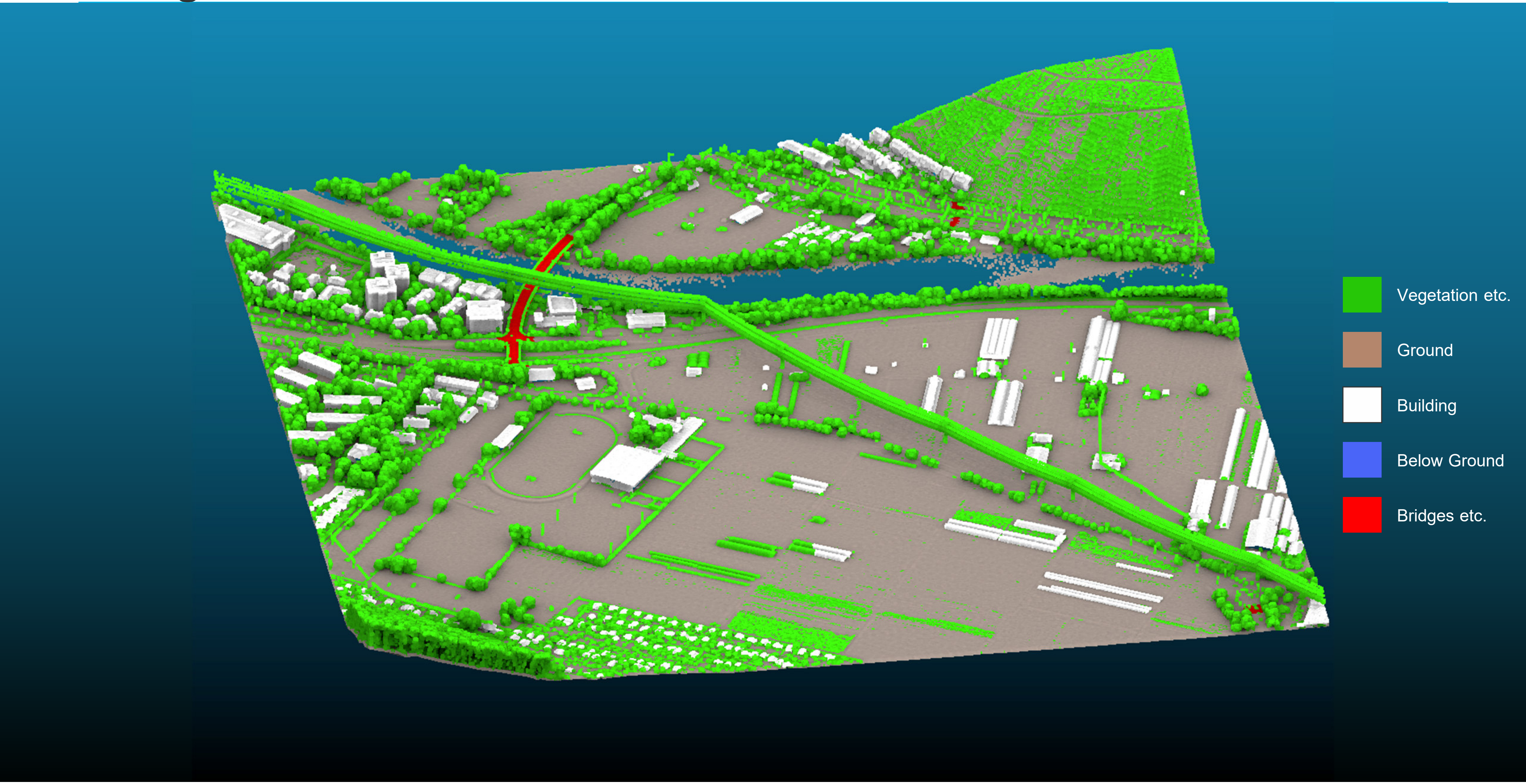
0,25 m

⇒ OA = **96%**

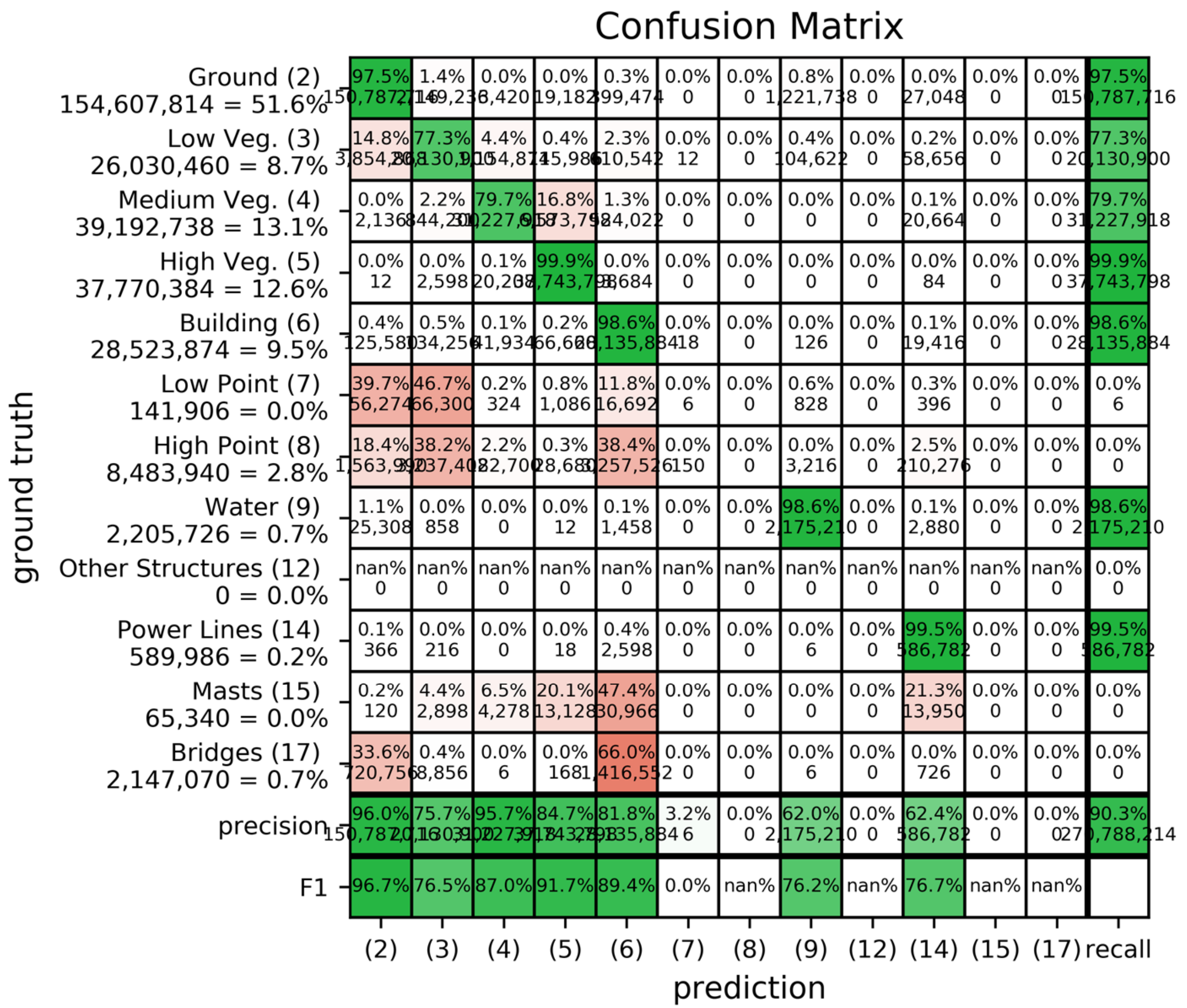


Vegetation etc.
Ground
Building
Water
Bridges etc.

Stuttgart – Ground Truth



Vorarlberg – Results (training set)



Vorarlberg – Results (validation set)

Confusion Matrix

ground truth \ prediction	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(12)	(14)	(15)	(17)	recall
Ground (2) 45,285,680 = 47.9%	95.0% 43,019,776	3.1% 1,389,732	0.0% 4,804	0.1% 50,756	1.3% 583,056	0.0% 0	0.0% 0	0.5% 228,152	0.0% 0	0.0% 9,404	0.0% 0	0.0% 0	95.0% 43,019,776
Low Veg. (3) 7,480,116 = 7.9%	13.8% 1,033,020	76.6% 5,729,056	4.5% 338,476	1.0% 76,344	3.6% 267,320	0.0% 16	0.0% 0	0.2% 12,712	0.0% 0	0.3% 23,172	0.0% 0	0.0% 0	76.6% 5,729,056
Medium Veg. (4) 10,205,468 = 10.8%	0.0% 632	3.6% 367,452	71.4% 7,285,432	21.6% 2,201,268	3.3% 336,968	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.1% 13,716	0.0% 0	0.0% 0	71.4% 7,285,432
High Veg. (5) 23,719,804 = 25.1%	0.0% 28	0.0% 1,244	0.0% 11,536	99.9% 23,699,060	0.0% 7,684	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 252	0.0% 0	0.0% 0	99.9% 23,699,060
Building (6) 6,426,636 = 6.8%	1.2% 76,868	1.1% 68,892	0.2% 15,736	0.2% 10,012	97.2% 6,249,360	0.0% 20	0.0% 0	0.0% 56	0.0% 0	0.1% 5,692	0.0% 0	0.0% 0	97.2% 6,249,360
Low Point (7) 24,444 = 0.0%	74.8% 18,272	19.4% 4,744	0.0% 0	0.7% 164	3.4% 840	0.0% 0	0.0% 0	1.7% 416	0.0% 0	0.0% 8	0.0% 0	0.0% 0	0.0% 0
High Point (8) 1,042,768 = 1.1%	17.1% 178,296	59.2% 617,424	2.4% 25,156	0.4% 3,836	17.9% 186,468	0.0% 68	0.0% 0	0.0% 268	0.0% 0	3.0% 31,252	0.0% 0	0.0% 0	0.0% 0
Water (9) 14,864 = 0.0%	17.7% 2,632	0.2% 28	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	82.0% 12,184	0.0% 0	0.1% 20	0.0% 0	0.0% 0	82.0% 12,184
Other Structures (12) 0 = 0.0%	nan% 0	nan% 0	nan% 0	nan% 0	nan% 0	nan% 0	nan% 0	nan% 0	nan% 0	nan% 0	nan% 0	nan% 0	0.0% 0
Power Lines (14) 163,204 = 0.2%	0.9% 1,536	0.1% 176	0.0% 0	0.1% 244	2.4% 3,948	0.0% 0	0.0% 0	0.9% 1,520	0.0% 0	95.5% 155,780	0.0% 0	0.0% 0	95.5% 155,780
Masts (15) 36,500 = 0.0%	0.0% 4	2.0% 736	3.0% 1,112	34.9% 12,736	49.9% 18,196	0.0% 0	0.0% 0	0.0% 0	0.0% 0	10.2% 3,716	0.0% 0	0.0% 0	0.0% 0
Bridges (17) 81,784 = 0.1%	33.5% 27,424	1.1% 932	0.0% 20	0.8% 648	64.5% 52,760	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0
precision	97.0% 43,019,776	70.0% 5,729,056	94.8% 7,285,432	91.0% 23,699,060	81.1% 6,249,360	0.0% 0	0.0% 0	4.8% 12,184	0.0% 0	64.1% 155,780	0.0% 0	0.0% 0	91.2% 86,150,648
F1	96.0%	73.2%	81.5%	95.2%	88.4%	nan%	nan%	9.0%	nan%	76.7%	nan%	nan%	