Deep Learning for Robust Normal Estimation in Unstructured Point Clouds

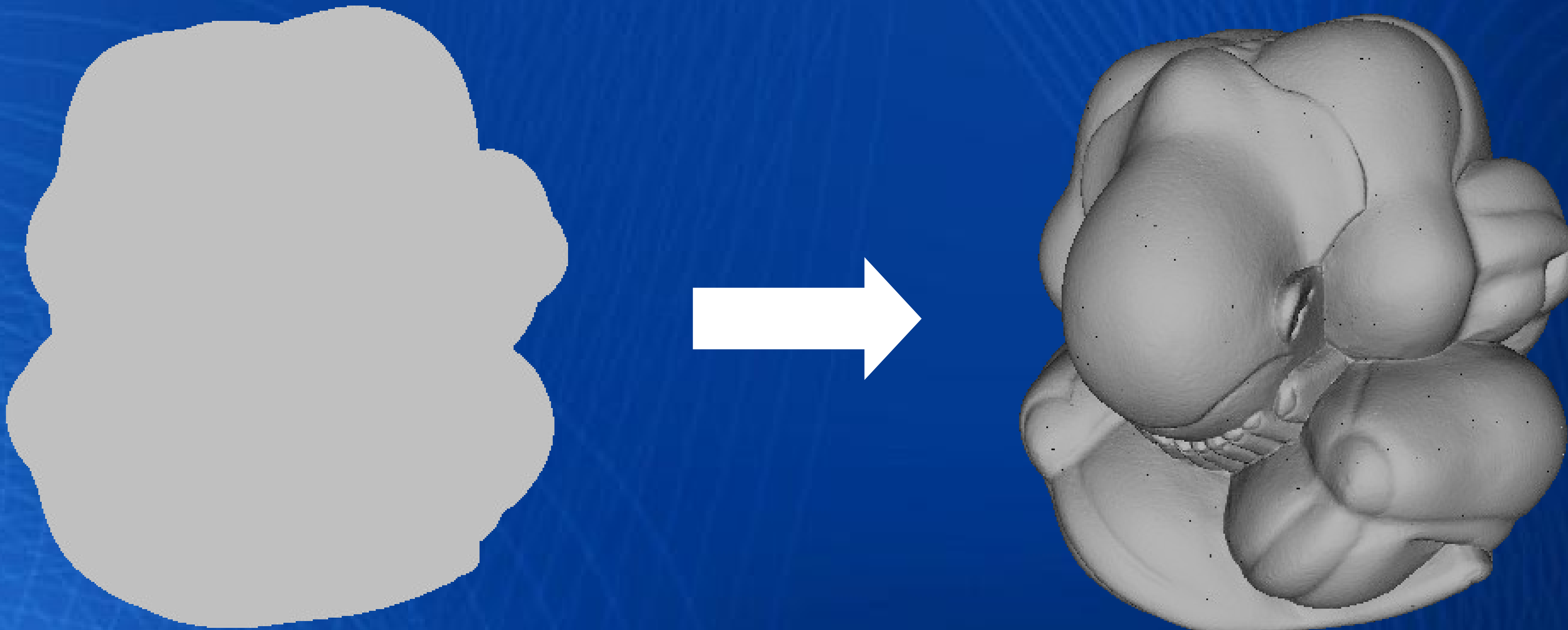
Alexandre Boulch

ONERA
THE FRENCH AEROSPACE LAB

Renaud Marlet

École des Ponts
ParisTech

Normal estimation in point clouds

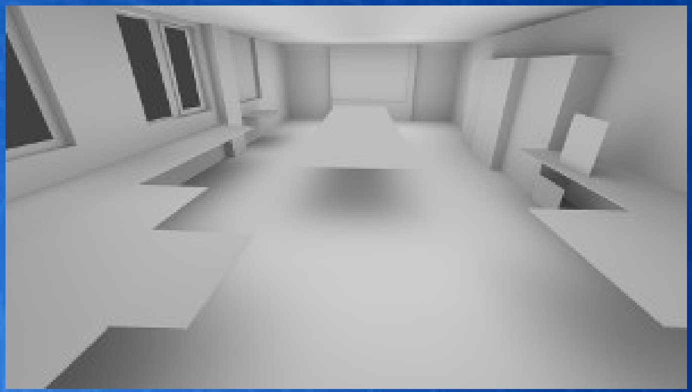


Normal: 3D normalized vector

At each point: local orientation of the surface

Normal estimation in point clouds

Primitive extraction [SWK07]



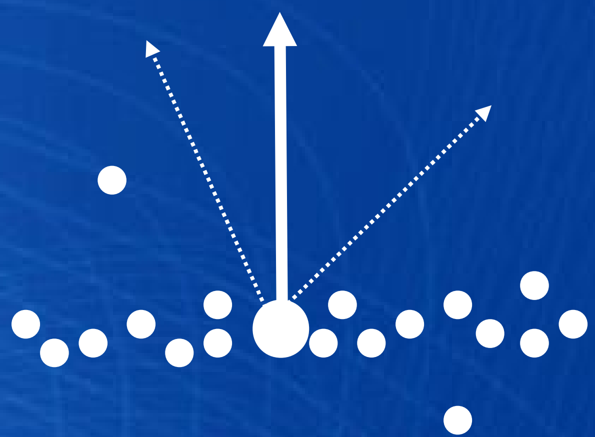
Surface reconstruction [BDLGM14]

Rendering [ABCO*03]

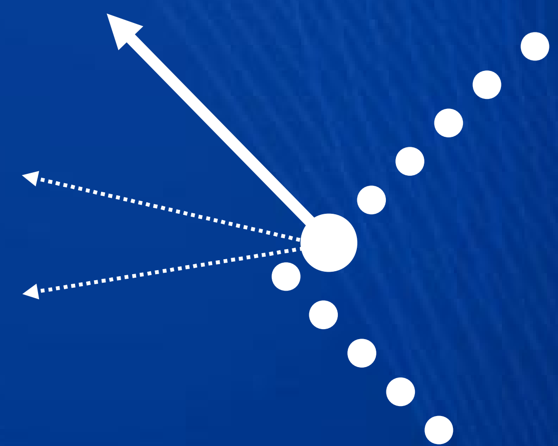
Full references in the paper

Normal estimation in point clouds

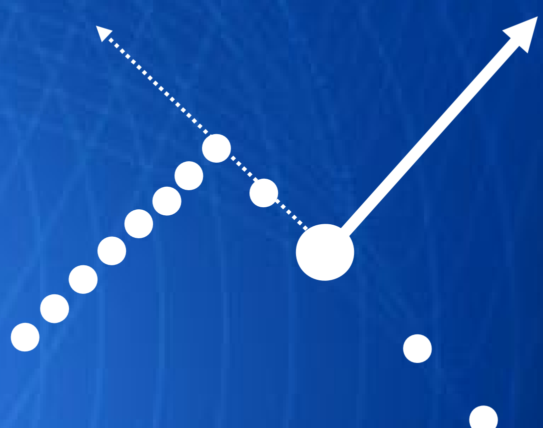
Main issues



Noise and outliers



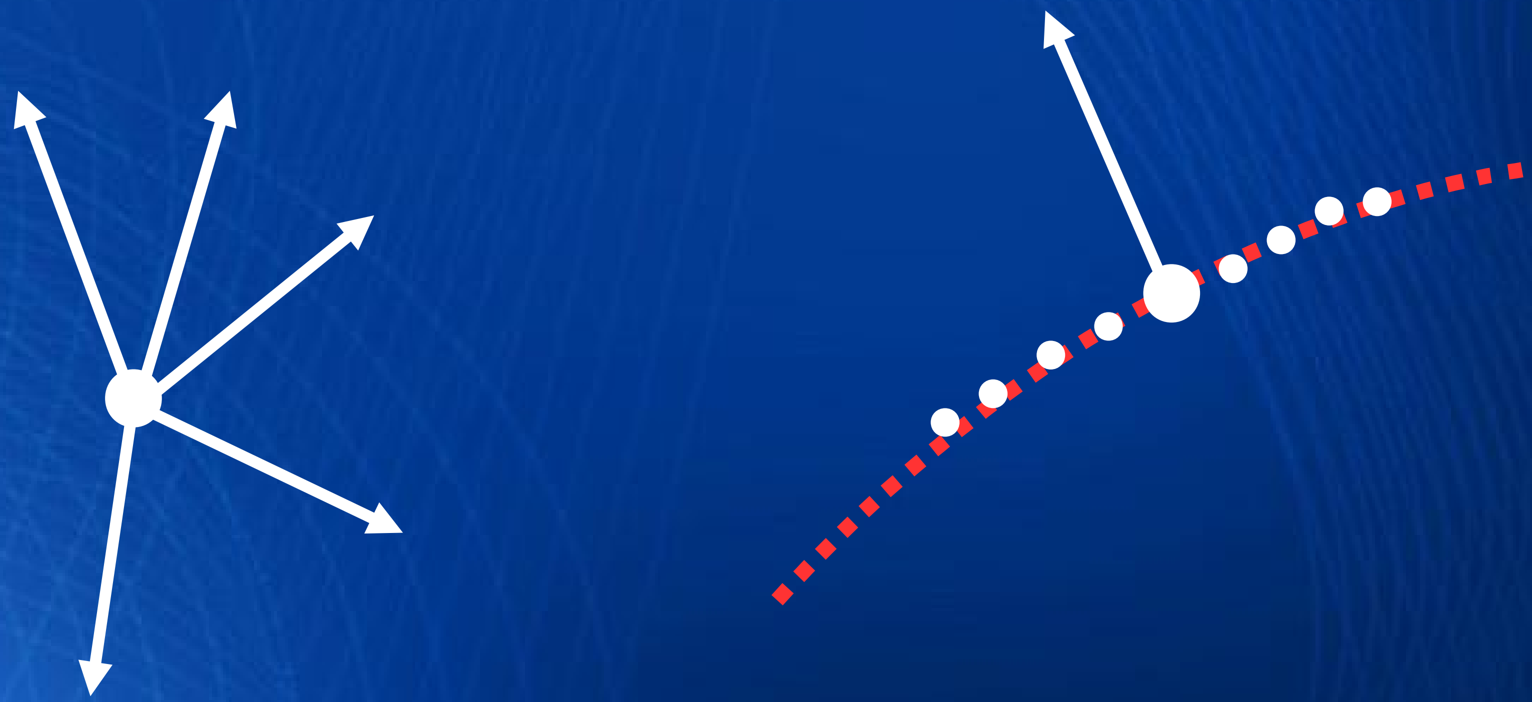
Sharp features



Density variations

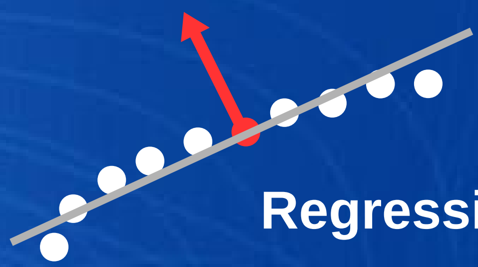
Computation time

Normal estimation in point clouds



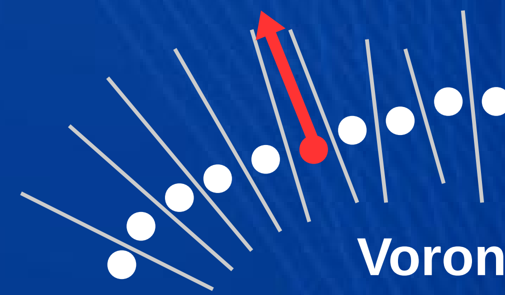
Normal estimation need neighborhood information

Existing methods



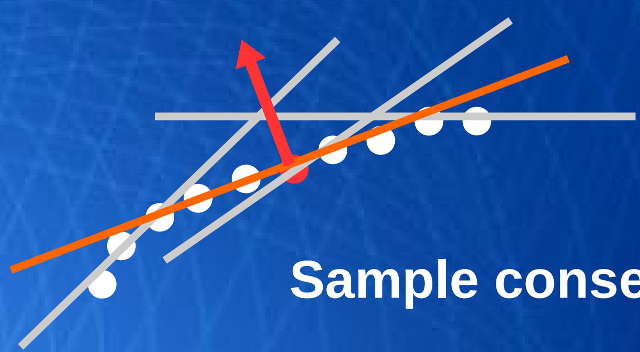
Regression

[HDD*92] HOPPE et al
[CP05] CAZALS and POUGET



Voronoï

[DG04] DEY and OSWAMI



Sample consensus

[LSK*10] LI et al.

Hough Transform

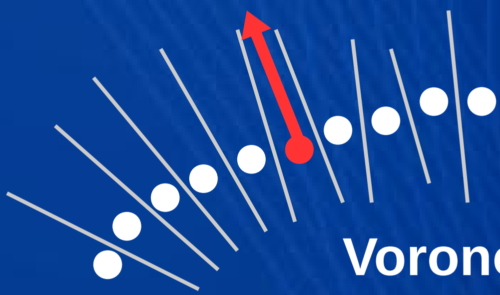
[BM12] BOULCH and MARLET

Other references in the paper

Existing methods

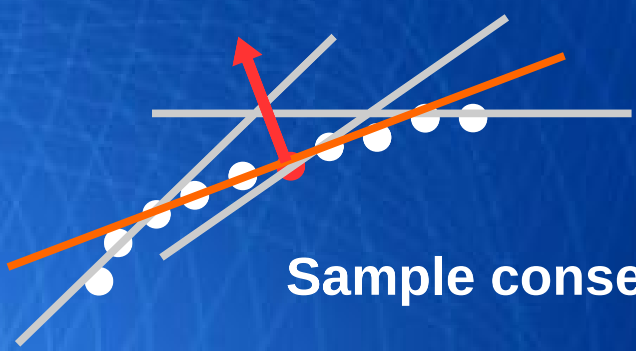
Regression

[HDD*92] HOPPE et al
[CP05] CAZALS and POUGET



Voronoï

[DG04] DEY and OSWAMI



Sample consensus

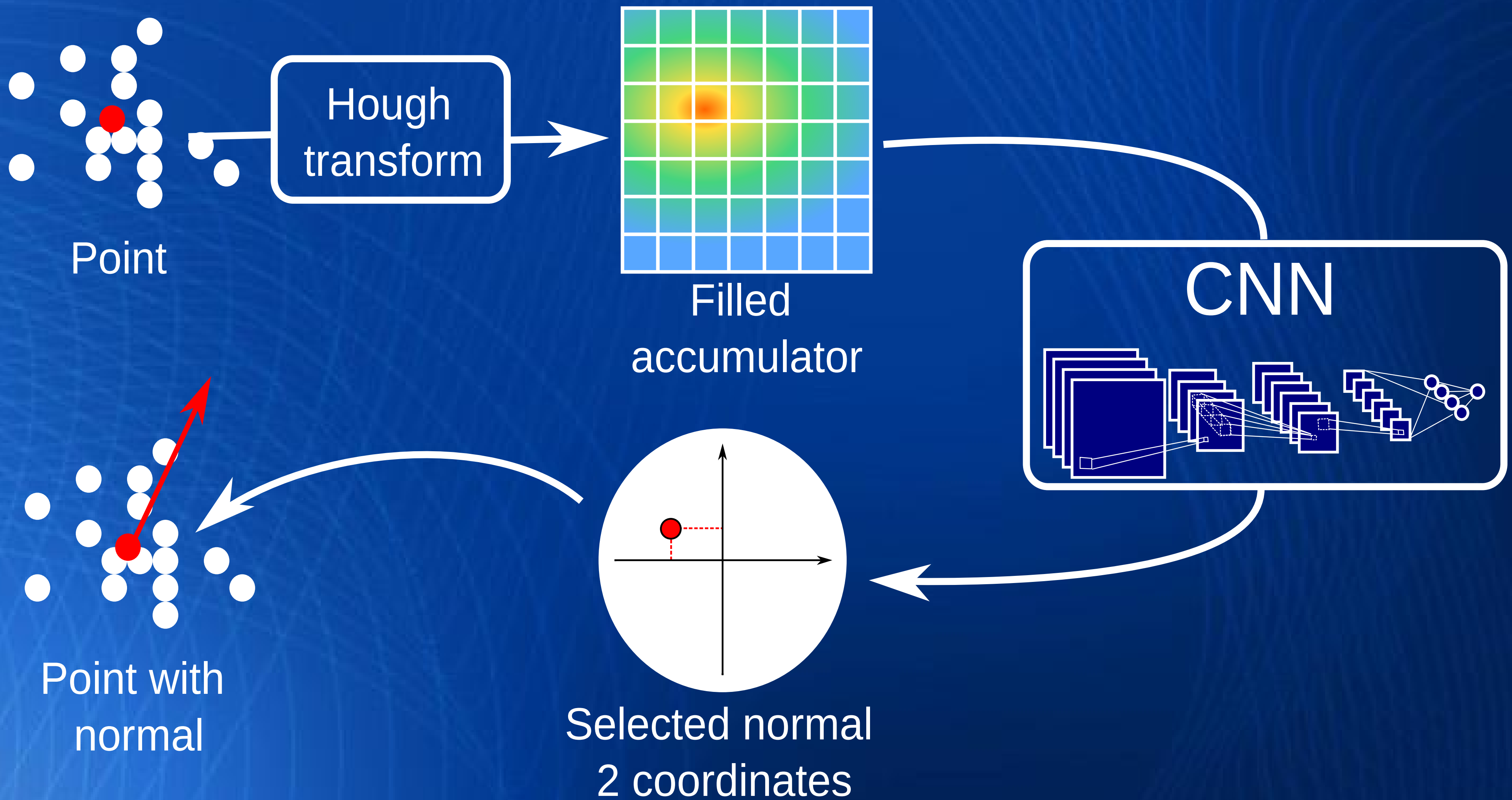
[LSK*10] LI et al.

Hough Transform

[BM12] BOULCH and MARLET

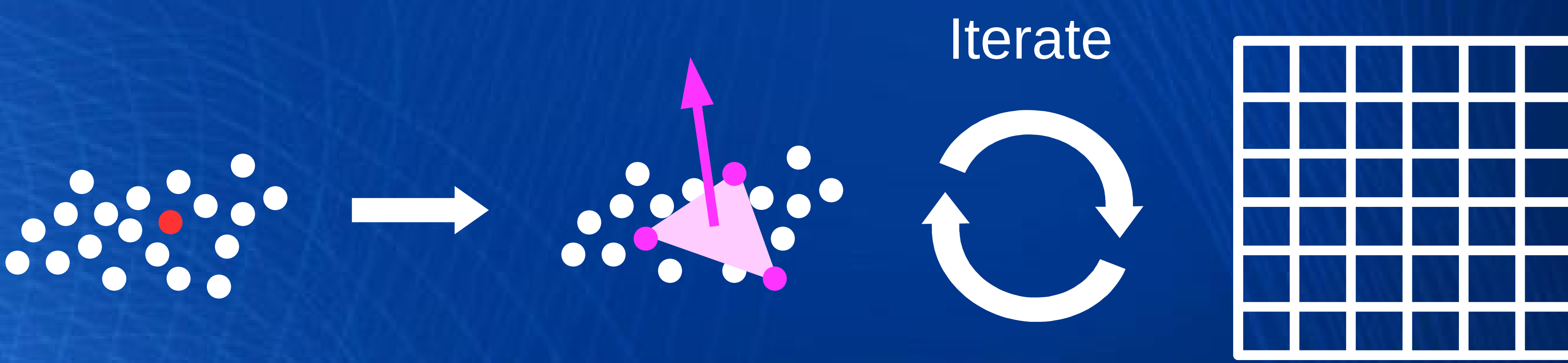
Other references in the paper

Principles of our method



Robust Randomized Hough Transform

Principle



Point and neighborhood

Pick three points
Estimate plane normal

Vote in the accumulator

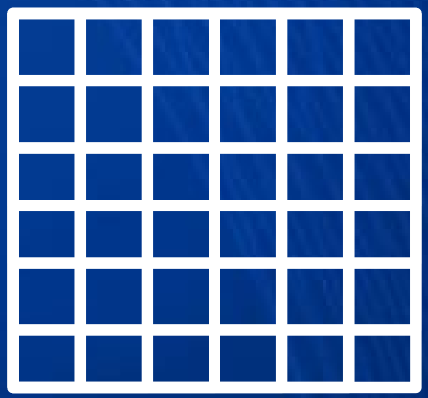
Statistical bounds on number of hypotheses to pick in the paper

Robust Randomized Hough Transform

Accumulator design

Discretized half sphere
[BM12]

Need for multiple
evaluations

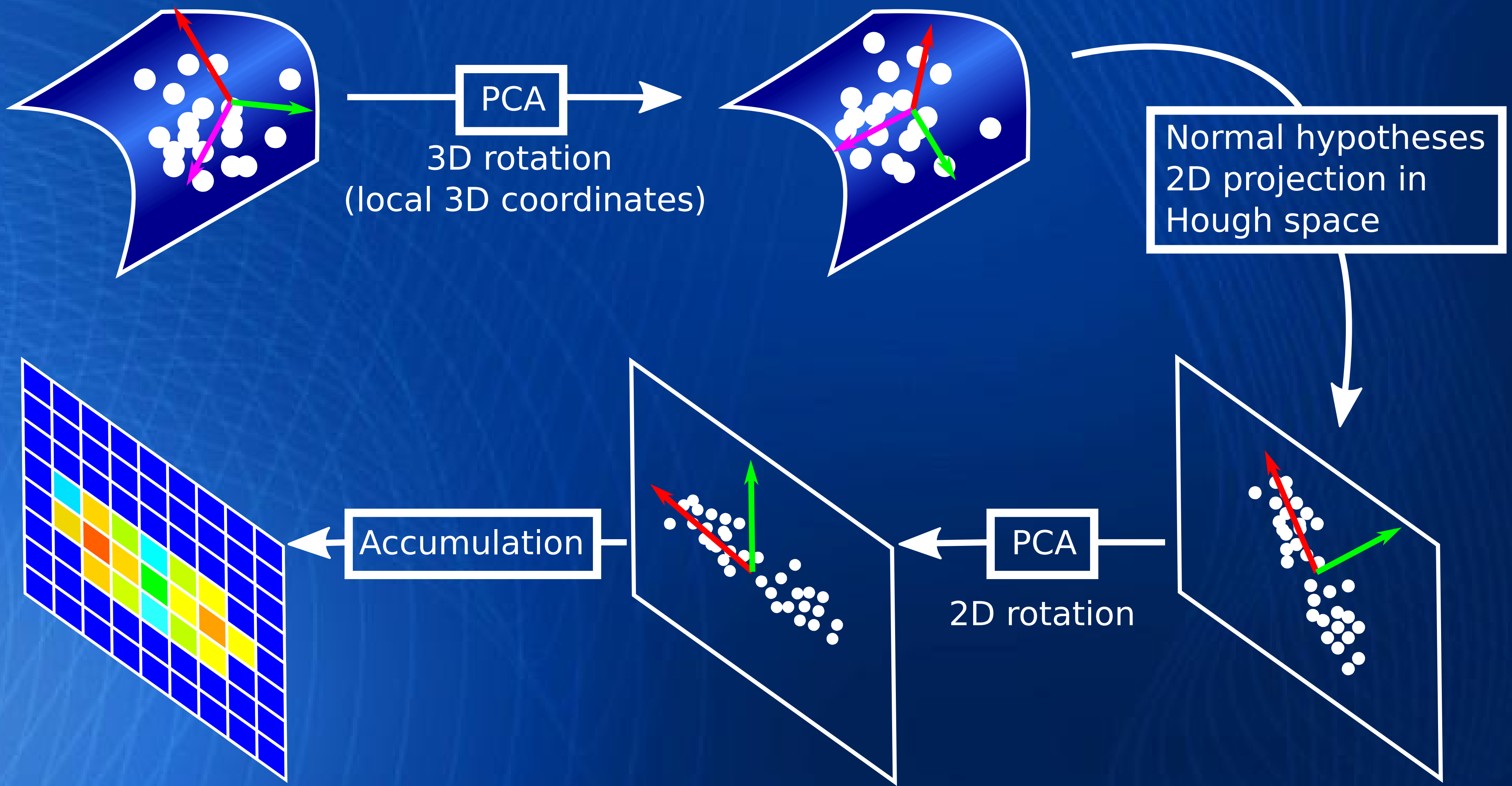


Grid accumulator

Better resolution
Suited for CNN

Robust Randomized Hough Transform

Accumulator design



Estimation from accumulator

Filled accumulator

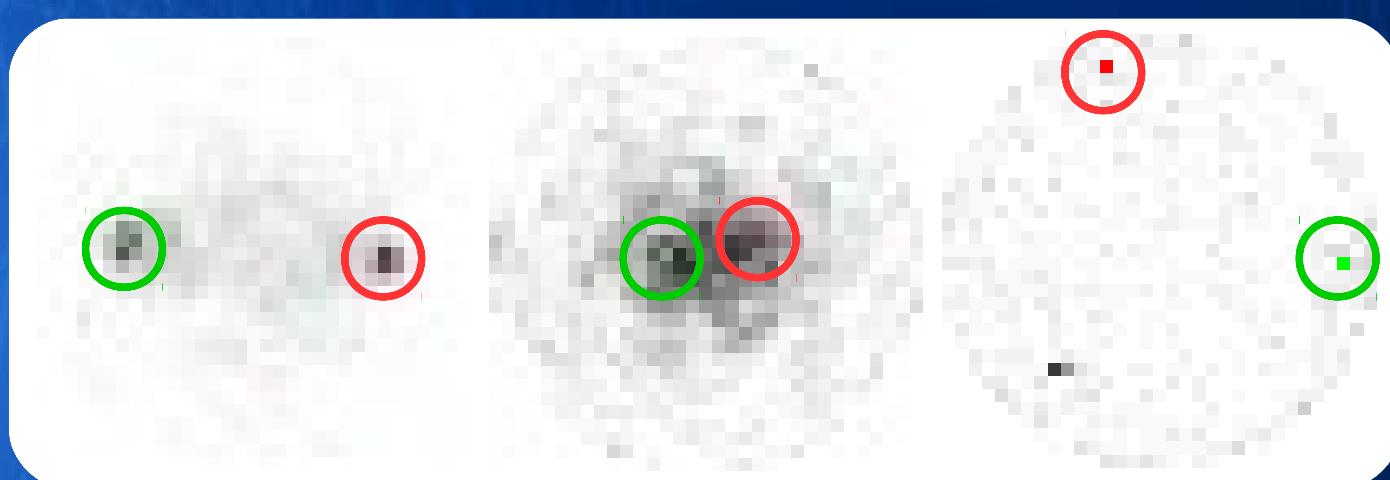
Normal coordinates in Hough Space

Estimation from accumulator

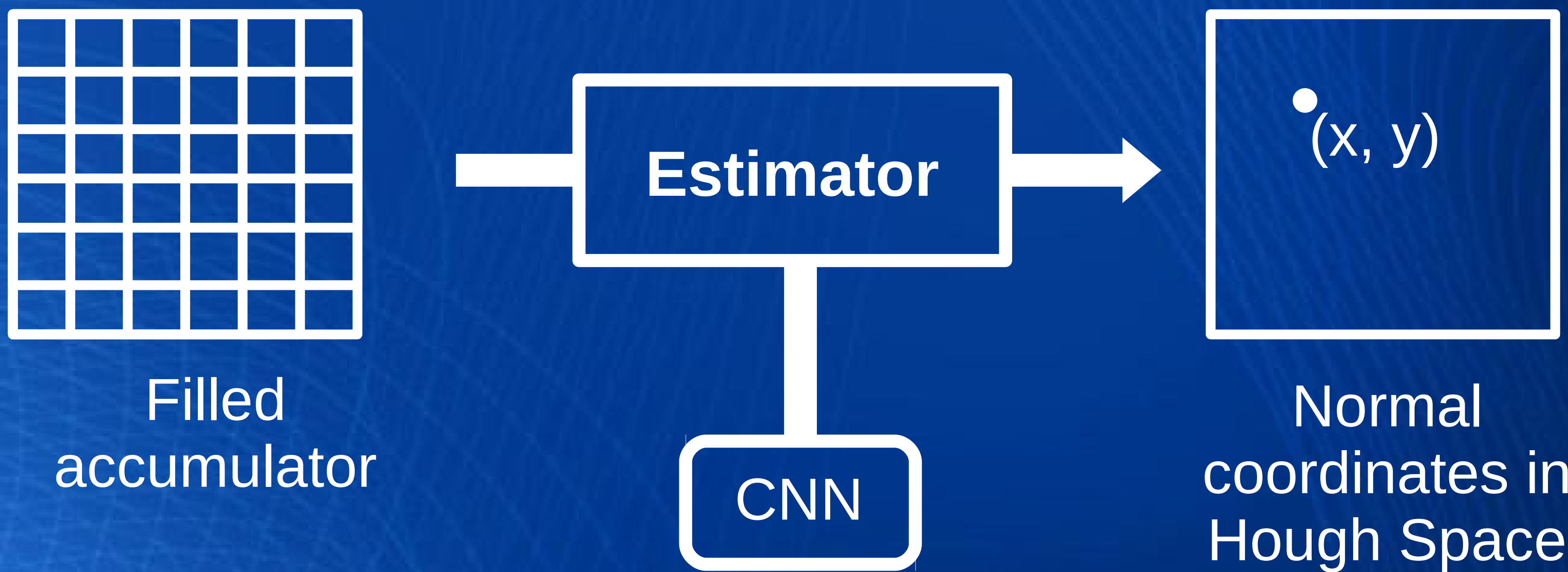
Filled accumulator

[BM12]
Maximum of the accumulator

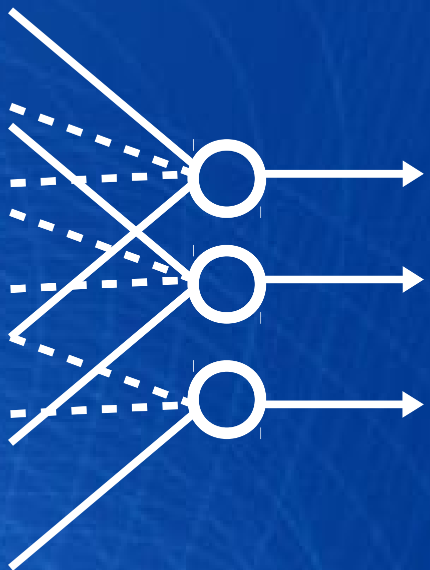
Normal coordinates in Hough Space



CNN for normal estimation



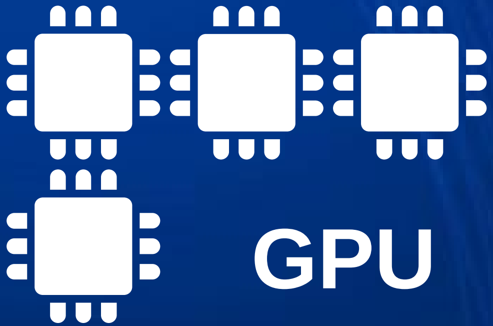
Deep learning



80's and 90's
Theory,
optimization...

+

DATA

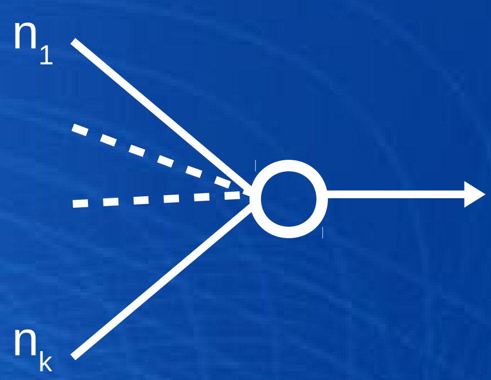


GPU

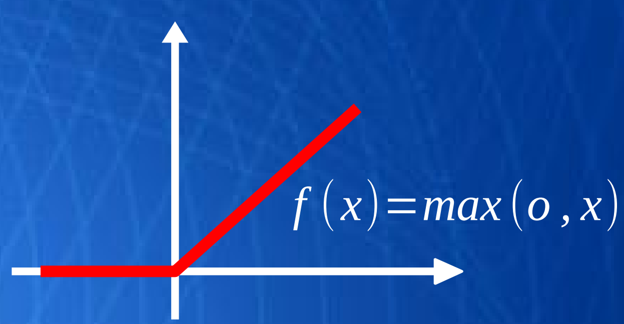
Since 2000's

Convolutional Neural Networks

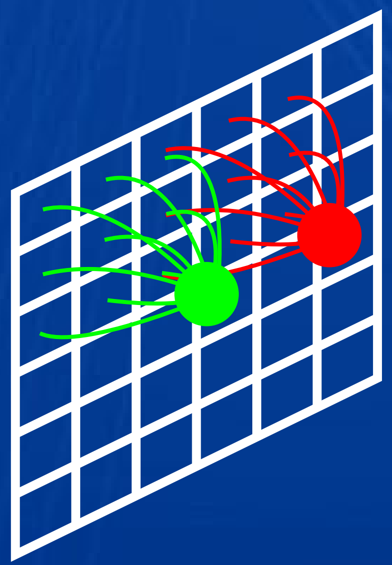
Main layer types



Fully connected
Input: all neurons
of previous layer

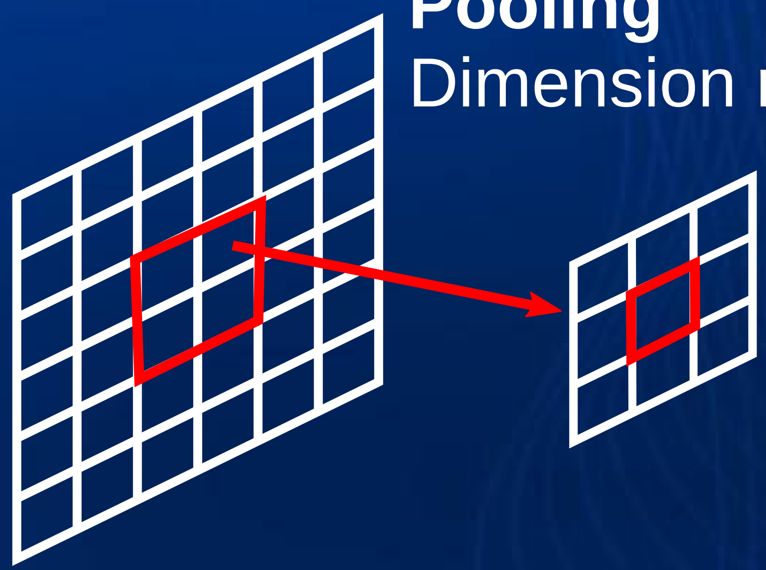


Activation layer
ReLU, Tanh
Increase non linearity



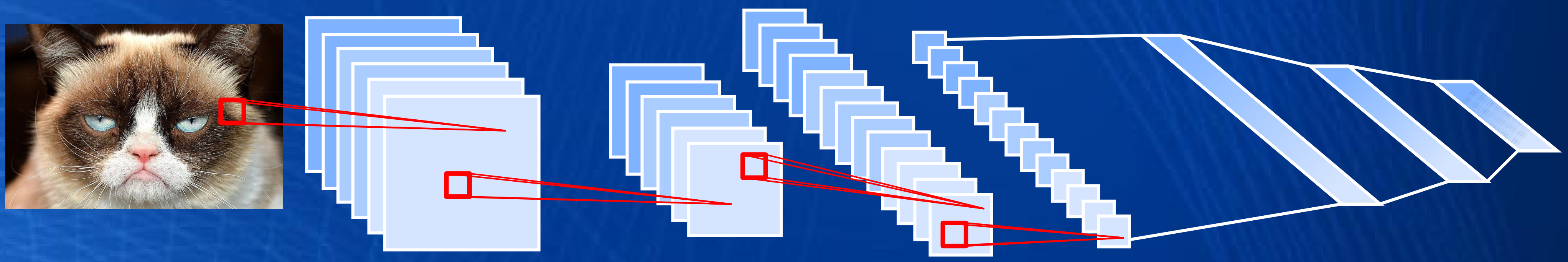
Convolutions
Input: rectangle of pixels
Regular grid of neurons
Share weights

Pooling
Dimension reduction



CNN for normal estimation

LeNet like architecture



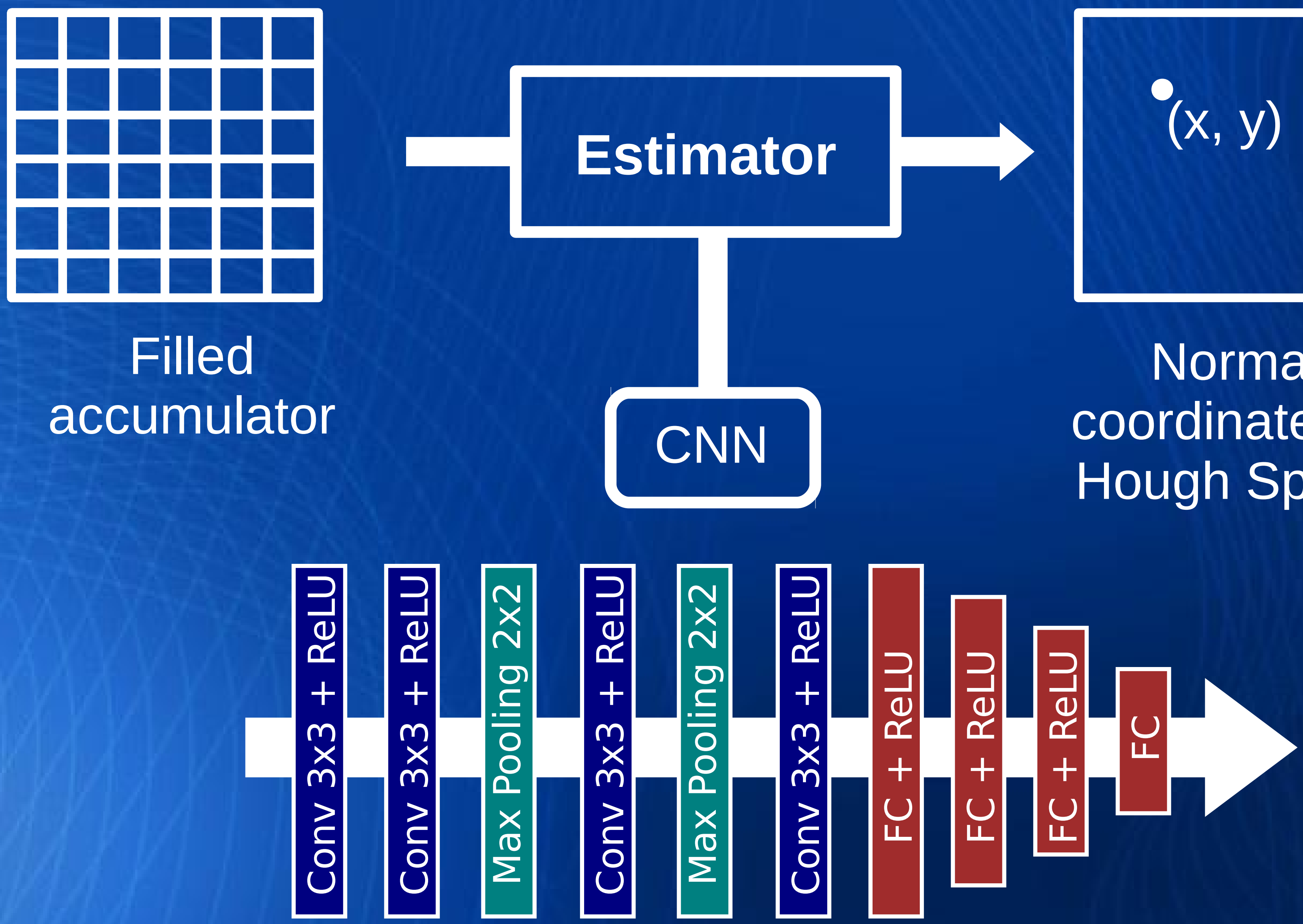
Descriptor

Stack of Convolutions and Pooling

Classifier

Stack of fully connected layers

CNN for normal estimation

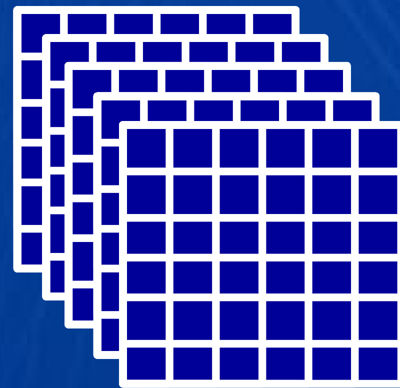


CNN for normal estimation

Training

Requires annotated data for training

Using big mesh:
- not satisfactory
- not enough difficult points
- real point clouds (no ground truth)



Training accumulators

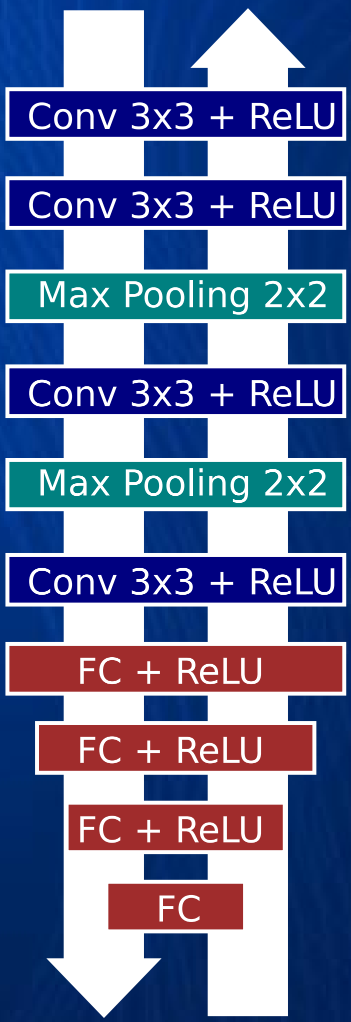
$n_1 = (x_1, y_1)$

$n_2 = (x_2, y_2)$

$n_3 = (x_3, y_3)$

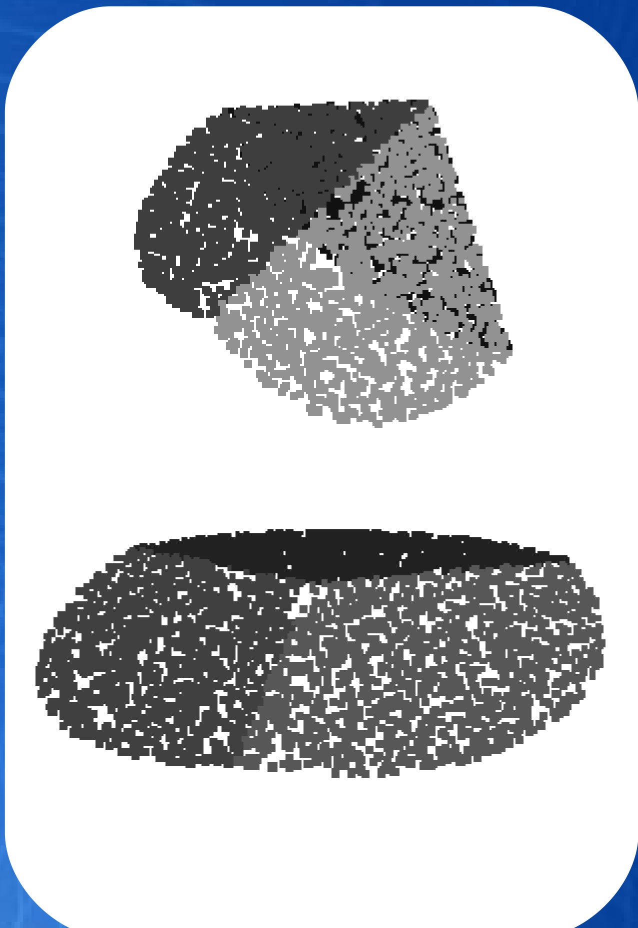
...

Corresponding normals

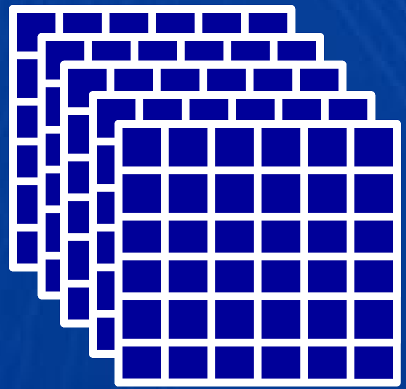


CNN for normal estimation

Training



Synthetic angles

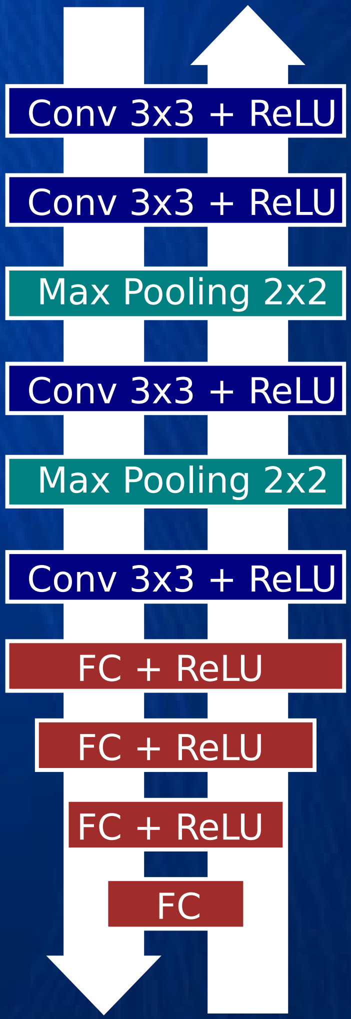


Training accumulators

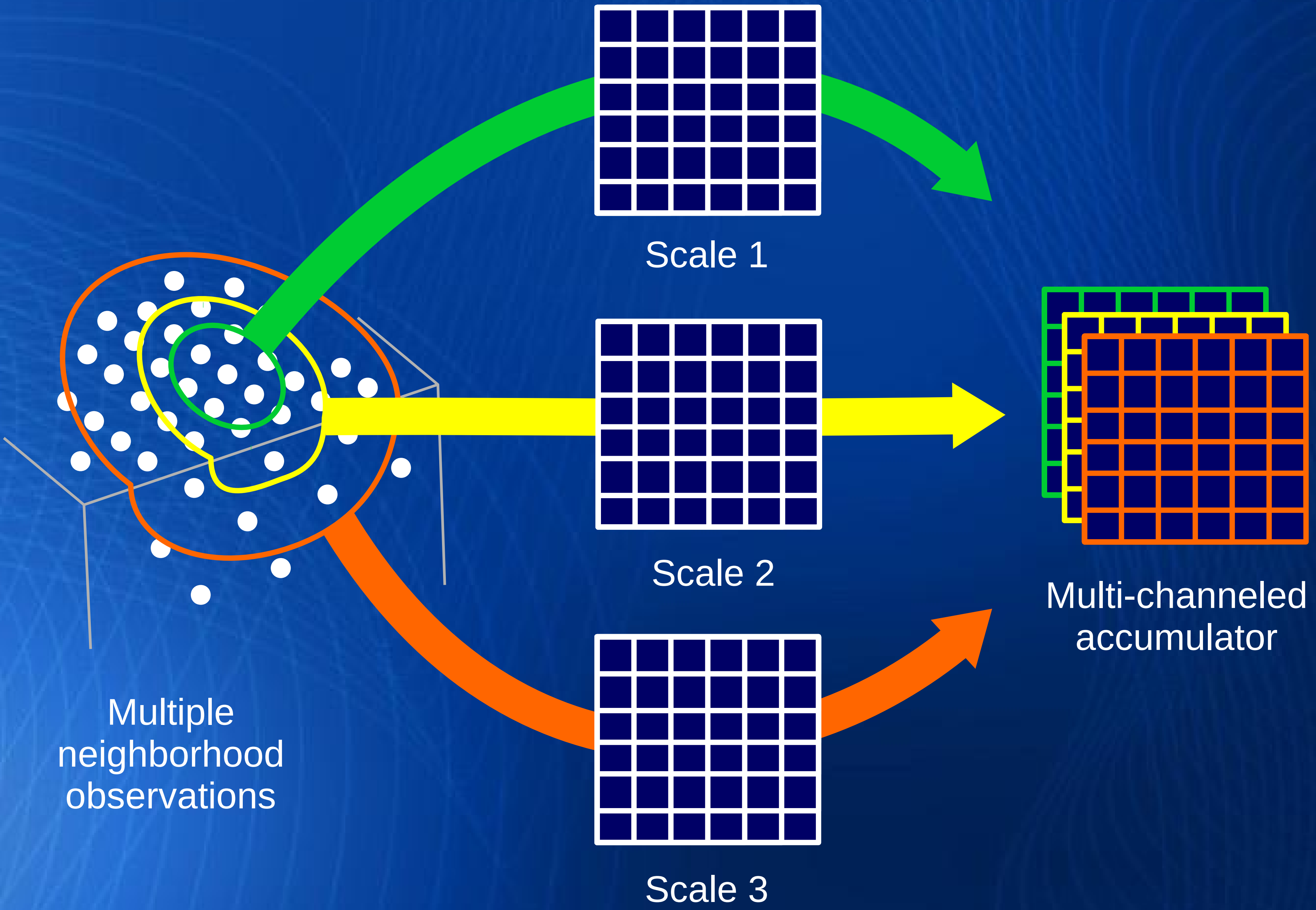
$$n_1 = (x_1, y_1)$$
$$n_2 = (x_2, y_2)$$
$$n_3 = (x_3, y_3)$$
...

Corresponding normals

$$L = \|\mathbf{n} - \hat{\mathbf{n}}\|_2^2$$

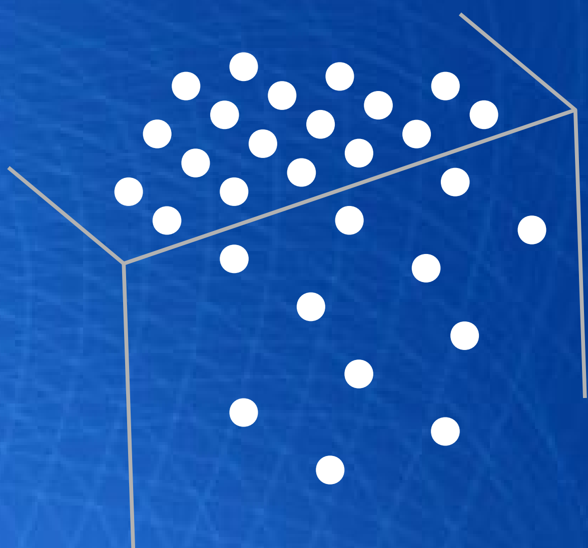


CNN for normal estimation

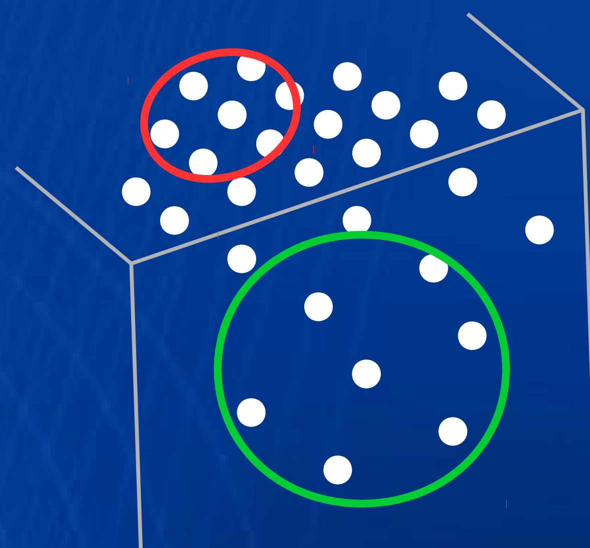


Robust Randomized Hough Transform

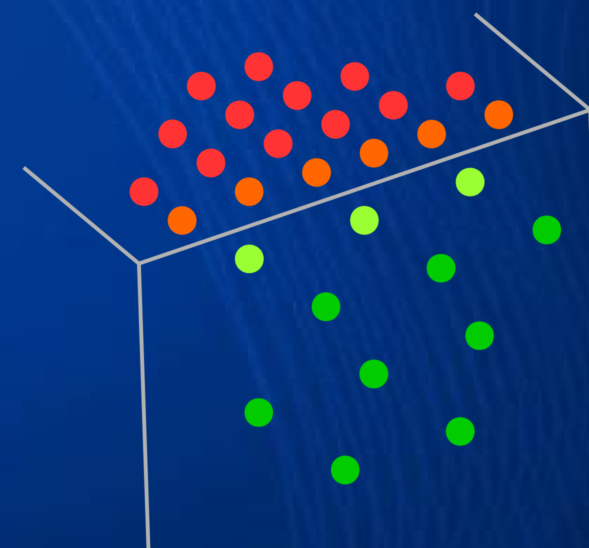
Robustness to density variation



Point cloud with density variations

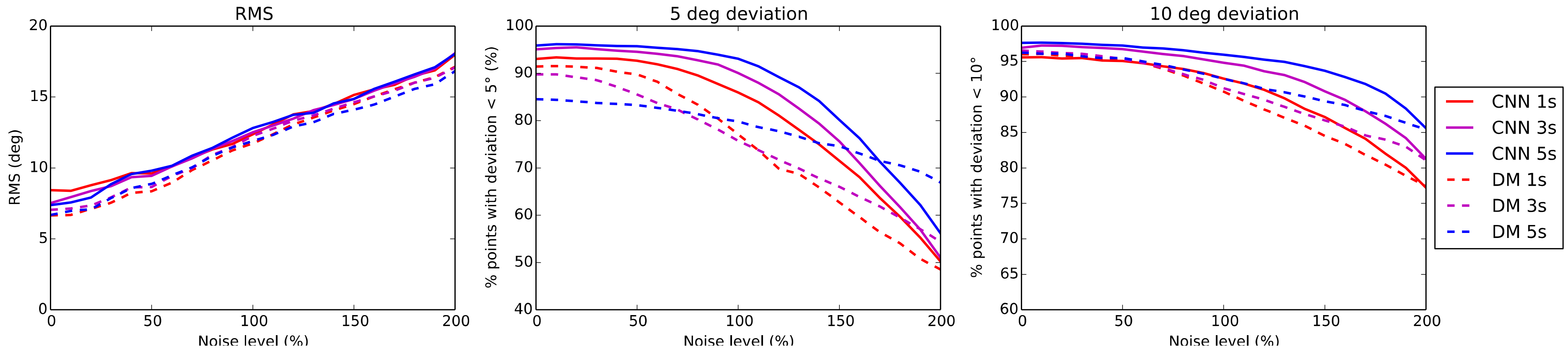


Compute local scale

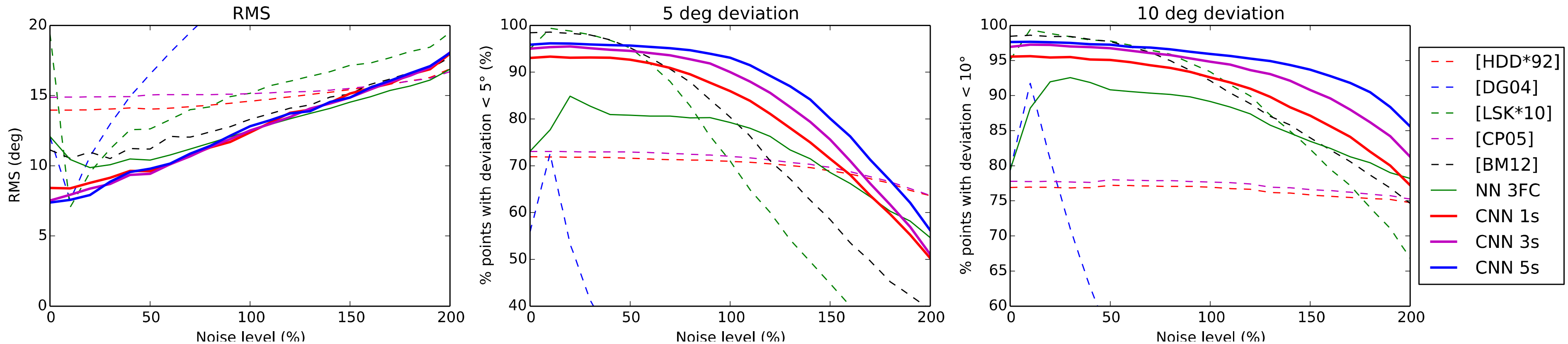


Assign picking probability to each point

Experiments

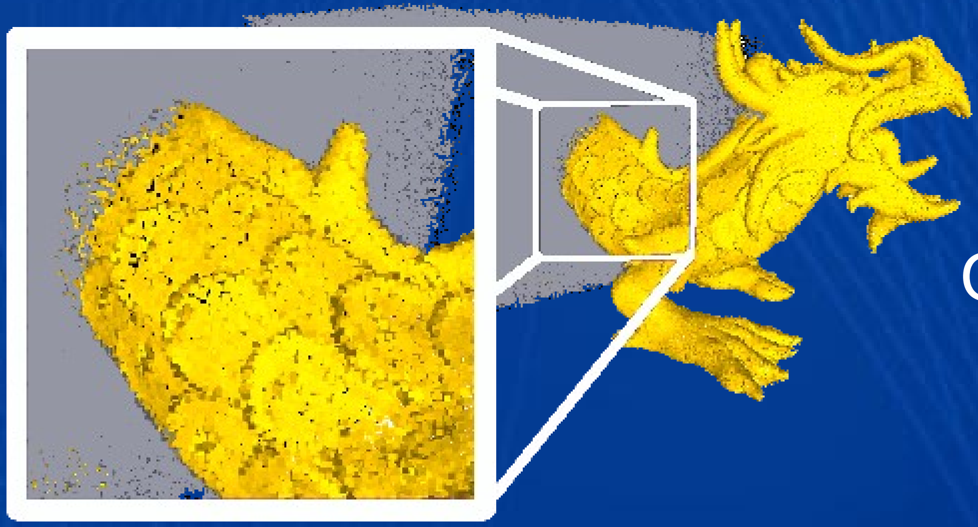


Depthmap + CNN vs Hough + CNN

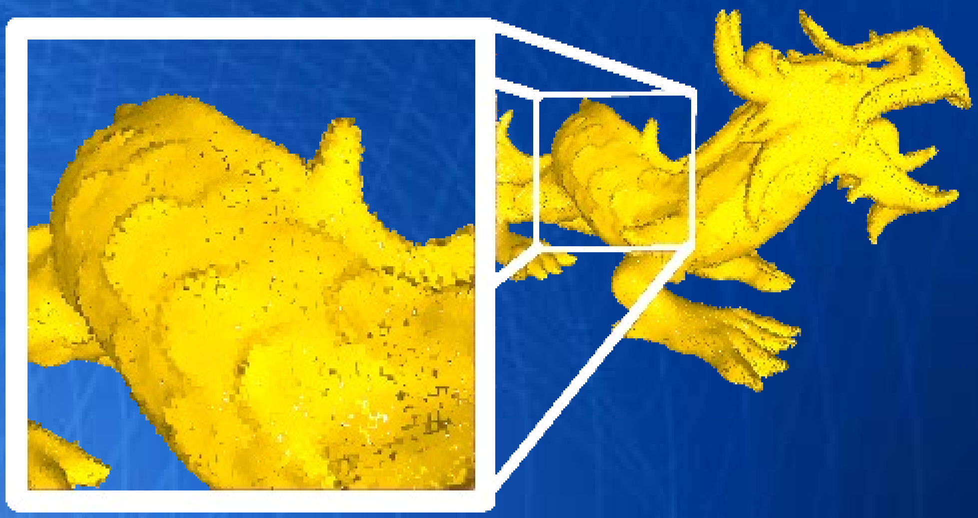


Comparison with existing methods

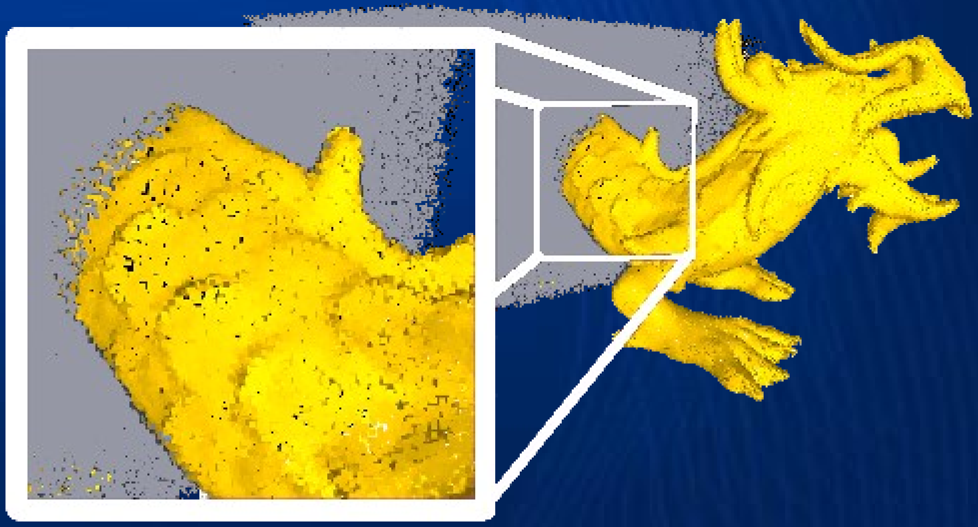
Experiments



Ground truth + outliers

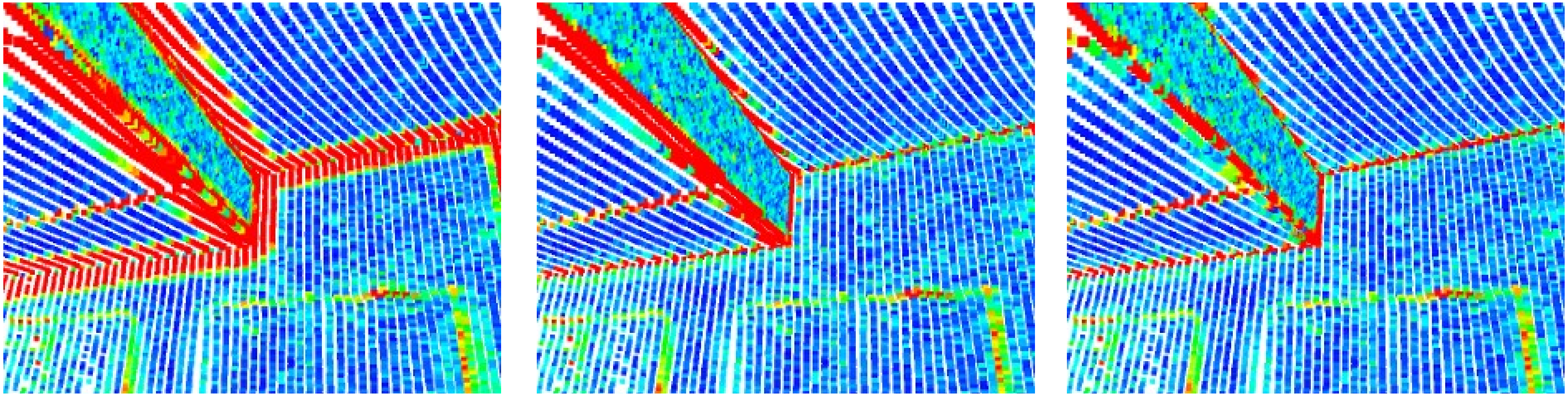


Hough + CNN
Estimation without outliers



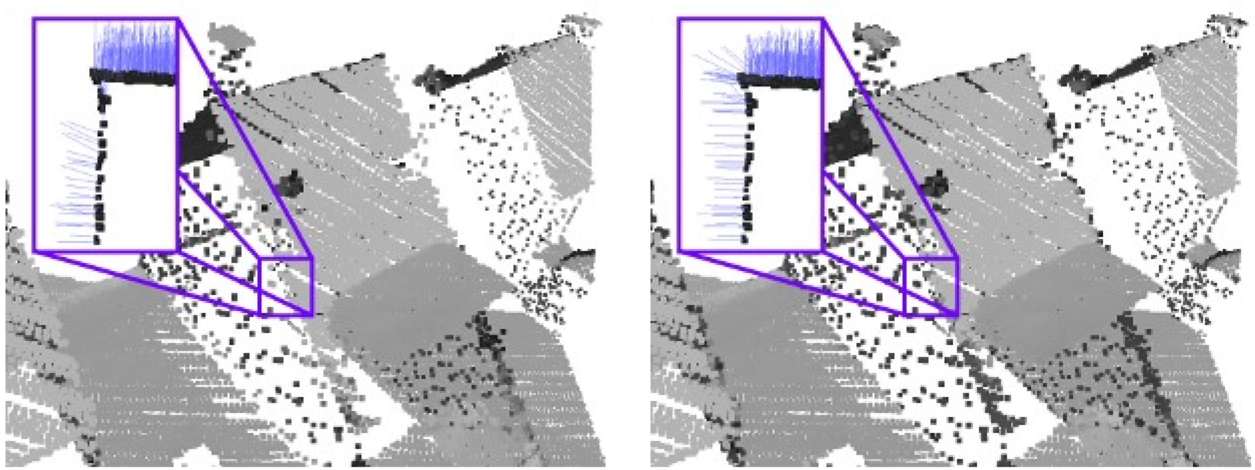
Hough + CNN
Estimation with outliers

Experiments



Plane fitting [HDD*92] | Our method plain | Our method density-adaptive

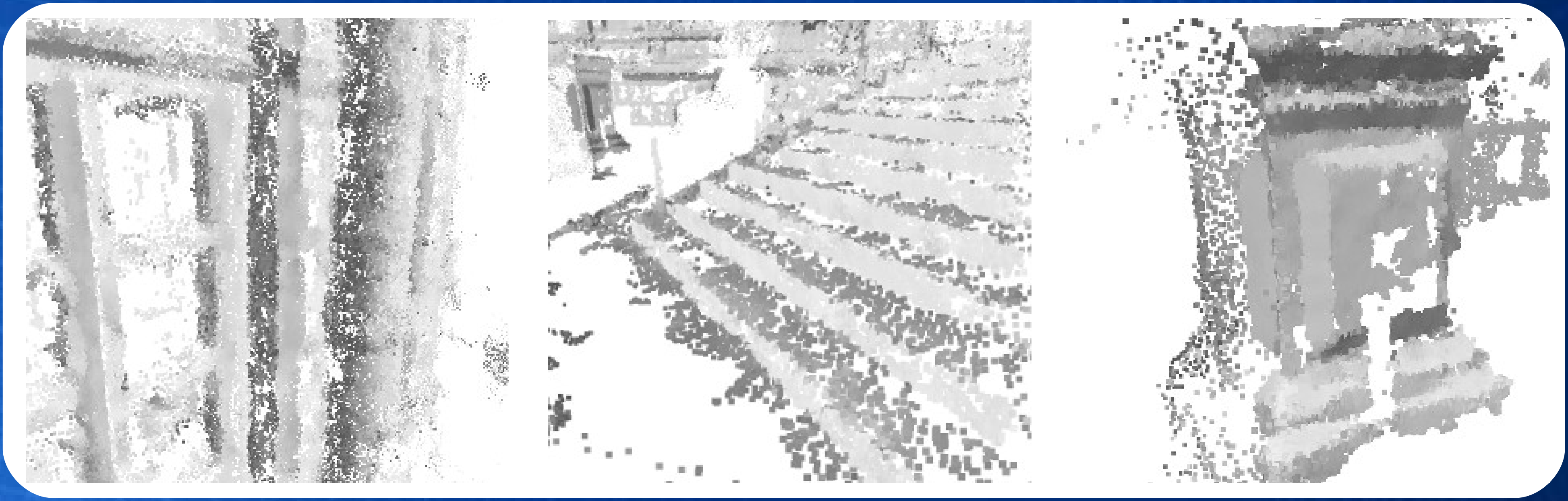
Ground laser office scene



Our method plain | Our method density-adaptive

Aerial laser scene (DFC 2015)

Experiments



Château de Sceaux, SfM point cloud

Reconstructed using OpenMVG by Pierre Moulon
https://github.com/openMVG/openMVG

When to use it ?

✘ Smooth surface
Homogeneous density

✔ Sharp edges
Density variations
Outliers
Noise

Perspectives

Adaptation to structured point clouds

Work on training sets and architectures

Geometric transformation in Hough space

Conclusion

Normal estimation in unstructured point cloud

New trend in geometry processing

Deep data driven approach

Hough transform helps the network

Thank you

</code>

sites.google.com/site/boulchalexandre

Hough + CNN :

C++ and Lua/Torch 7

Trained models available

[BM12] Header only

Original: CGAL, PCL

Updated version (density sensitive): Eigen / NanoFlann

Computation times

Model	Cube	Armadillo	DFC Detail	Omotondo	DFC tile
Size	20k	173k	185k	997k	2.3M
[HDD*92]	0.3	2.1	1.9	12	25
[DG04]	3.2	55	41	441	1243
[CP05]	5.8	50	54	304	711
[BM12]	1.9	13	11	44	147
[LSK*10]	8.8	64	75	392	902
CNN 1s	**4.5**	**33**	**34**	**183**	**423**
CNN 3s	**5.9**	**48**	**52**	**273**	**639**
CNN 5s	**7.9**	**69**	**73**	**382**	**897**