Master in Computer Vision
Barcelona

Module M4: 3D Vision

Lecture 4.10: **Point Cloud Processing and Applications**

Lecturer: Josep R. Casas

Outlook

Session 9b: 3D Sensors and 3D data

– Motivation & Principles

  Image vs Range sensing

– 3D Sensors

– **3D data** → concepts, problems tools

Session 10: 3D processing & applications

– Pointcloud processing:

  PCL (2011) & Open 3D (2018)

  Organized/Unorganized processing

  Applications

3D data

Depth, cloud, mesh...

3D data representation

- Depth map
  - **2.5D** (concept: RGBD = 2D + depth)

- Point cloud
  - **organized**: keeps relationships in sensor neighborhood
  - **unorganized**: one can ***just*** compute nearest neighbors in 3D

- Mesh
  - nice scanned/reconstructed surfaces: watertight / convex...

Point cloud data

- ## Organized point-cloud

  Resemble an organized image (or matrix-like) structure, with data split into rows and columns (data from stereo, depth or TOF sensors)

  → **projectable** point cloud: has a correlation according to a pinhole camera model between the (u, v) index of a point in the organized point cloud and the actual 3D values (x, y, z).
  This correlation can be expressed as: $u = f*x/z$ and $v = f*y/z$

  → knowing the relationship between adjacent points (e.g. pixels), **nearest neighbor operations** are much more efficient, thus speeding up the computation and lowering the costs of certain algorithms in **PCL**

- ## Unorganized point-cloud

  Non-regular sampling of 3D space
  Neighborhood operations require ***KD tree*** search!

[Rusu 2009, Rusu 2011] PCL: pointclouds.org/documentation

[Friedman 1977], Wikipedia: wikipedia.org/wiki/K-d_tree

You might wander…

Why 3D data (3D objects/mesh) is state of the art in Graphics…
… whereas it is not so for 3D data (pointclouds) in analysis?

- Synthetic vs Captured!
- Perfectly located points vs captured (randomly distributed)
- Nice 3D animation vs real life

In addition

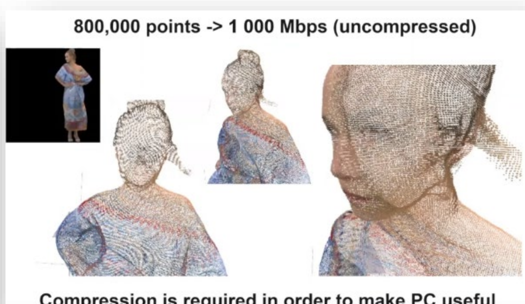
- CV is swiftly progressing towards industrial integration (image & video)
- Will pointclouds follow *easily* along? (point clouds & *stream* data)

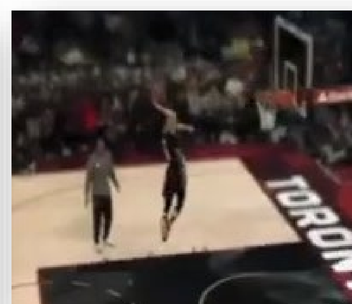
First problem: data management & transfer!

PointCloud

- unordered set of 3D points
- no relations among them
- points defined by
  - (x, y, z) … floats
  - (RGB or YUV) … 3x bytes
  - possibly reflectance, transparency…

1-3 Gbps/object!





M. Preda, [Point Cloud Compression in MPEG](), What's New in MPEG, Results from 131st Meeting, Jul 2020

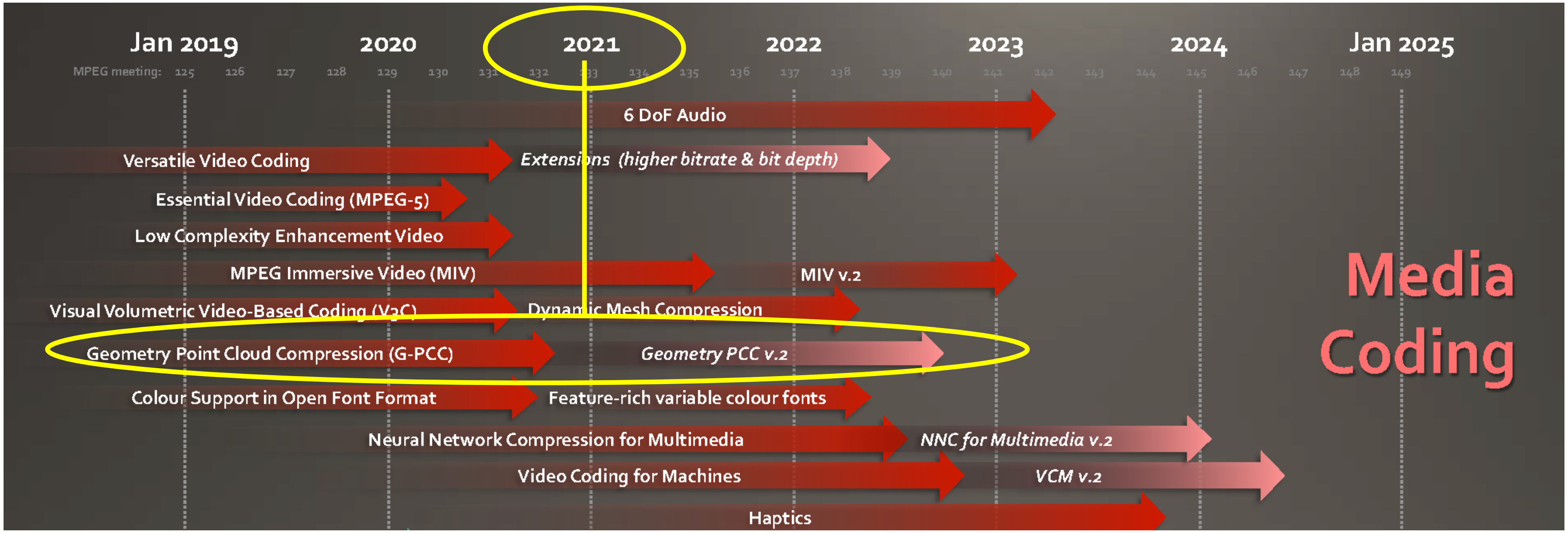
MPEG PCC roadmap

Point Cloud Compression early stages

similar series than (MPEG-1 > MPEG-2 > AVC > HEVC > VVC)

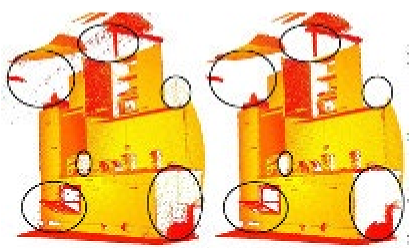
Implementation problems even in local installations

...local network, buses, memory

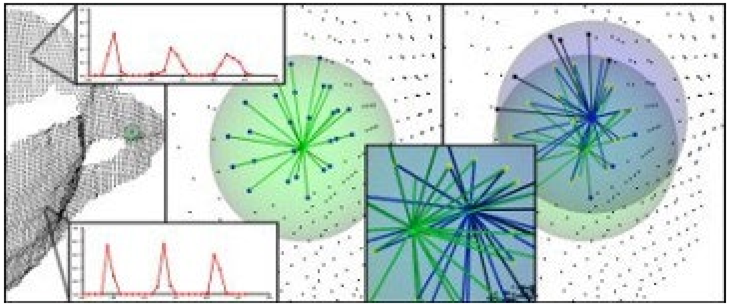


Point Cloud Library (PCL)

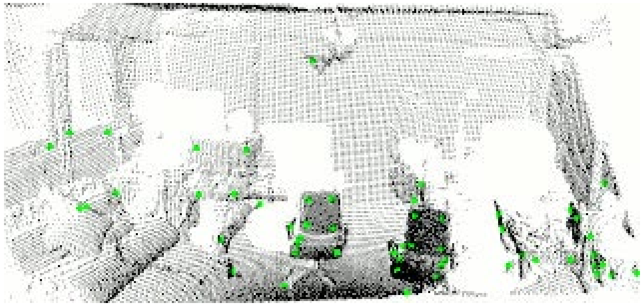
- New trend to process raw data produced by scanners
  - PCL library B.D Rusu (since 2011)
    pointclouds.org v1.11.1 (Jul 2020)
    github.com/PointCloudLibrary



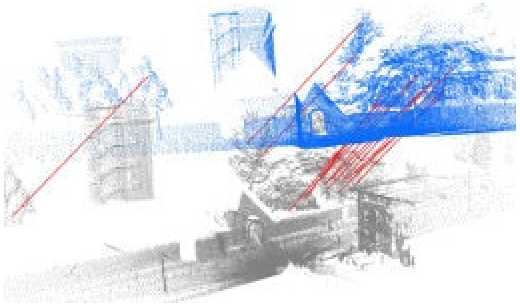
Filters



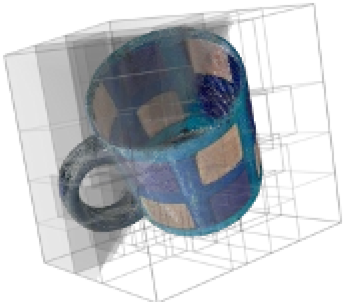
Features



Key points

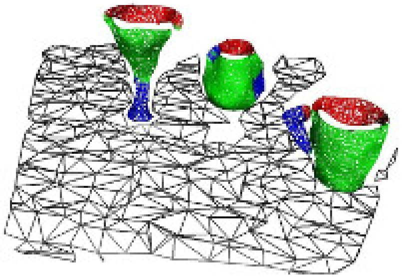


Segmentation



RANSAC

Octree



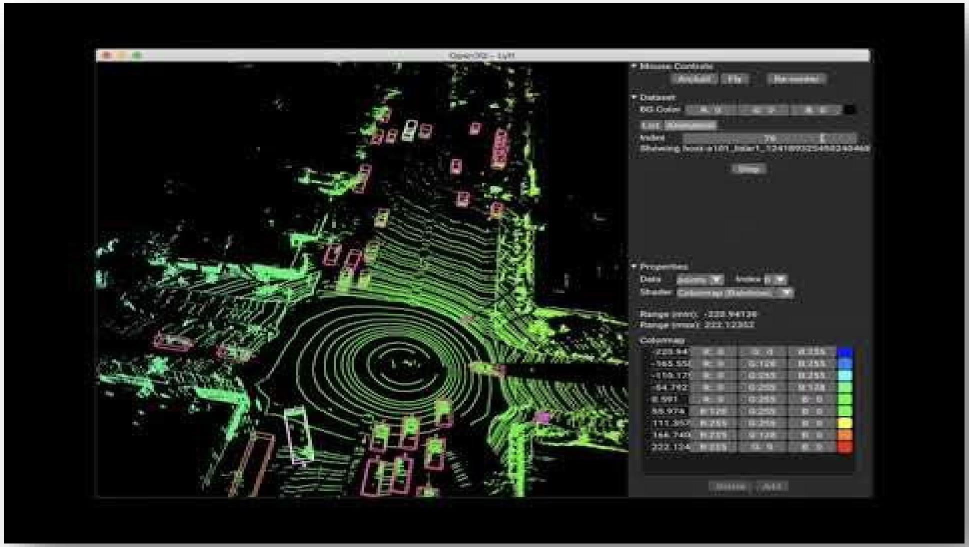
Visualization

Open3D

- A Modern Library for 3D Data Processing

Open3D open3d.org
github.com/intel-isl/Open3D
C++ or Python, Includes PCL

Intel ISL (since 2018)
v0.14.0 (Dec 2021)

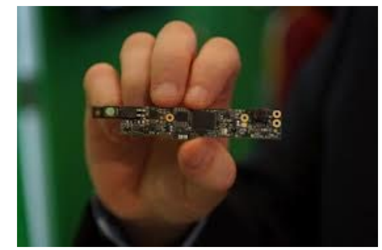


Xavier Suau, Human body analysis using depth data

ORGANIZED POINT CLOUD PROCESSING

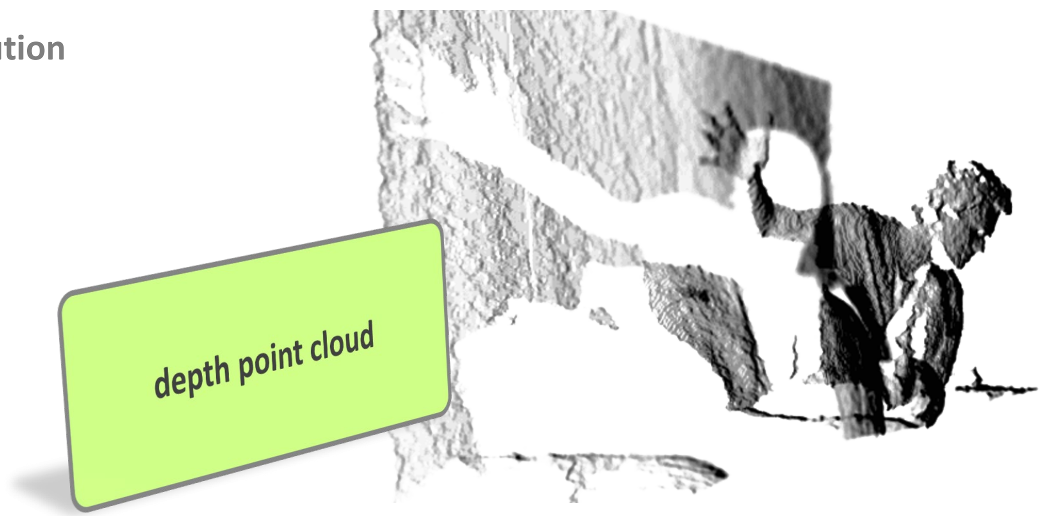
Depth, as if 3D were just images...

Human Body Analysis using Depth Data



Big
Expensive
Low-resolution

Small
~150€
high-resolution



Depth map / Point Cloud processing

Objective: To obtain information from depth camera frames

Depth Data

- Poor texture
- Sharp edges
- Low resolution

?

2D descriptors

- SIFT
- SURF
- Freak
- …

3D descriptors

- 3DSC [1]
- VFH [2]
- SHOT [3]
- …

OK

Other proposals

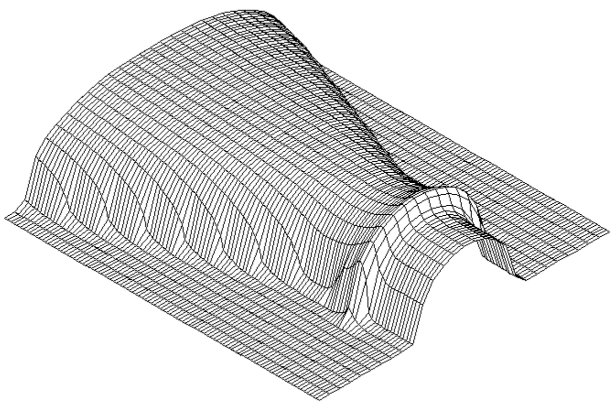
- **ORD [4]**
- R-NBLS

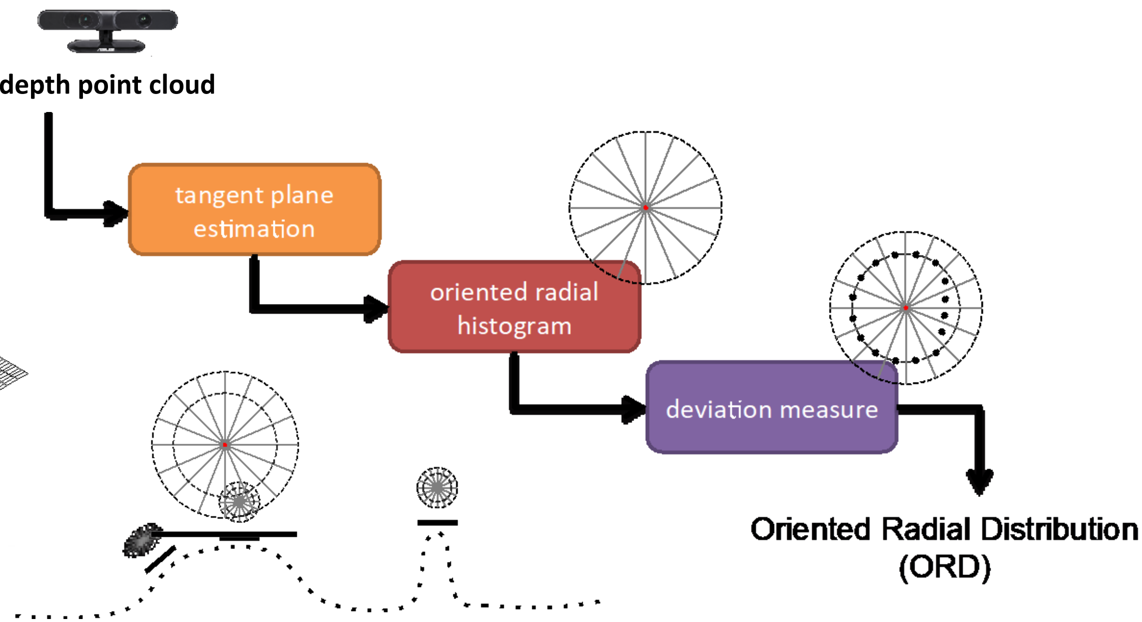
[1] Andrea Frome et al. Recognizing objects in range data using regional point descriptors. ECCV 2004
[2] R.B. Rusu, et al. Fast 3d recognition and pose using the viewpoint feature histogram. IROS 2010
[3] F. Tombari, et al. Unique signatures of histograms for local surface description. ECCV 2010
[4] X. Suau, et al, "Oriented Radial Distribution on Depth Data: Application to the Detection of End-Effectors," ICASSP 2012

Oriented Radial Distribution

Objective: Detect prominent and flat zones of a depth point cloud

depth point cloud
(simplified example)





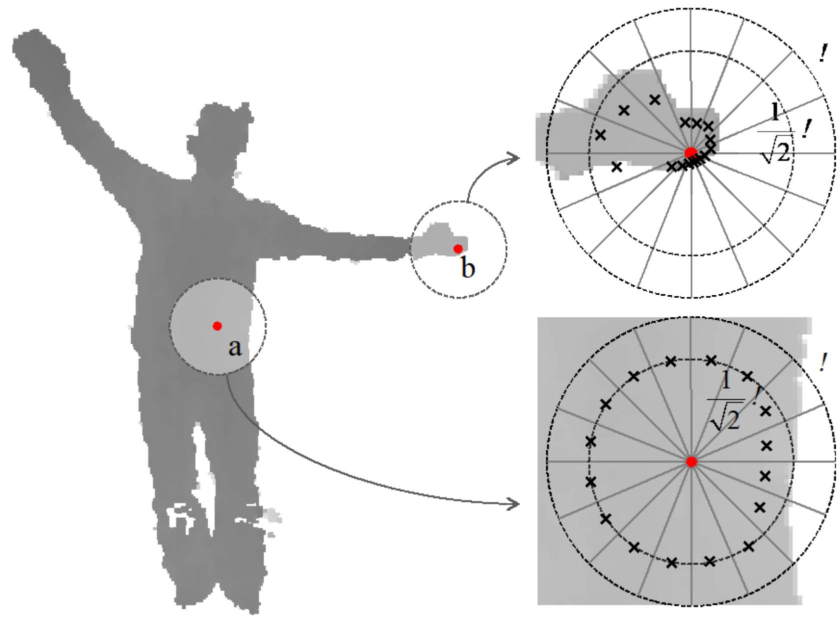
[4] X. Suau, et al, “Oriented Radial Distribution on Depth Data: Application to the Detection of End-Effectors,” ICASSP 2012

Oriented Radial Distribution

Objective: Detect prominent and flat zones of a depth point cloud

ORD Characteristics

- **Oriented** to the surface normals
- **Local** computation (neighborhood of a point)
- **Multiscale** (disk radius)
- Output: **histogram** or **scalar**



Deviation measure:

$$\Theta(\mathbf{z},\Omega,\xi) = \frac{1}{\frac{1}{\sqrt{2}}\rho K_f}\sum_{j=0}^{K_f}\left(\bar{\delta}_j - \frac{1}{\sqrt{2}}\rho\right)$$

a : low deviation → **FLAT**

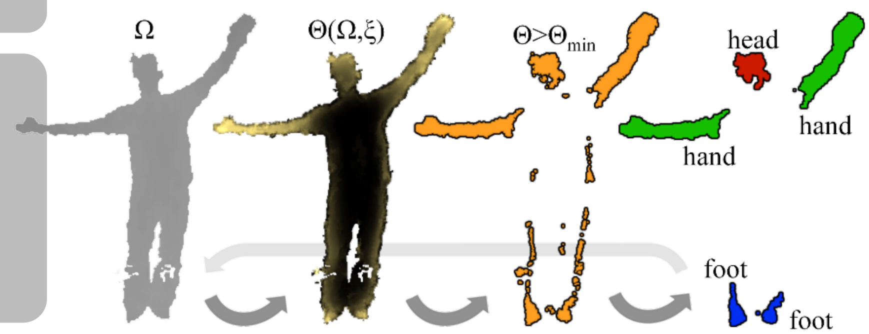
b : high deviation → **PROMINENT**

Oriented Radial Distribution

Classification of prominent zones

We propose to use probabilistic descriptors

- **Y :** Position, relative height of zone
- **S :** Size, area of the zone
- **A :** Shape, PCA aspect ratio of zone



Statistical moments of the descriptors

λ_k	$\mu_{\lambda_k}^{head}$	$\sigma_{\lambda_k}^{head}$	$\mu_{\lambda_k}^{hand}$	$\sigma_{\lambda_k}^{hand}$	$\mu_{\lambda_k}^{foot}$	$\sigma_{\lambda_k}^{foot}$
Y	62.18	7.48	29.43	29.06	−71.31	10.89
S	58.58	10.00	64.24	24.89	46.68	10.75
A	0.58	0.17	0.11	0.13	0.41	0.19

A blob **B** is classified

γ_i=**{head,hand,foot,**nothing**}**

depending on its

combined probability

$$
\begin{aligned}
P(\mathcal{B}=\gamma_i) &= P\Big((Y_B=\gamma_i)\wedge(S_B=\gamma_i)\wedge(A_B=\gamma_i)\Big)\\
&= f_Y^{\gamma_i}(\mathcal{B})\cdot f_S^{\gamma_i}(\mathcal{B})\cdot f_A^{\gamma_i}(\mathcal{B})
\end{aligned}
$$

$$
\text{with PDF:}\quad f_{\lambda_k}^{\gamma_i}(\mathcal{B}) = \frac{1}{\sigma_{\lambda_k}^{\gamma_i}\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{\lambda_k(\mathcal{B})-\mu_{\lambda_k}^{\gamma_i}}{\sigma_{\lambda_k}^{\gamma_i}}\right)^2}
$$

Other Point Feature Representations

- **Signature of Histograms of OrienTations (SHOT)**
- **VFH signatures**
- **Point Feature Histograms (PFH)**
- **Fast Point Feature Histograms (FPFH)**
  - See tutorials in PCL: pointclouds.org/documentation/tutorials

Point feature representations

A good point feature representation distinguishes itself from a bad one, by being able to capture the same local surface characteristics in the presence of:

- rigid transformations - 3D rotations and translations in the data should not influence the resultant feature vector estimation
- varying sampling density - a local surface patch sampled more or less densely should have the same feature vector signature
- noise - the point feature representation must retain the same or very similar values in the presence of mild noise in the data.

R. B. Rusu, "Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments," PhD TUM 2009

Martin Matilla, Alignment of 3D Point Clouds and RPS Detection

POINT CLOUD PROCESSING IN AUTOMOTIVE INDUSTRY

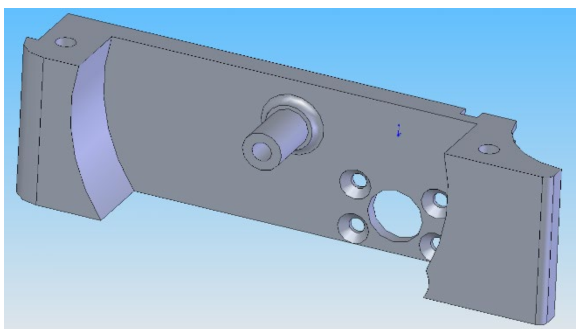
Point clouds (static), getting ready for the real thing...

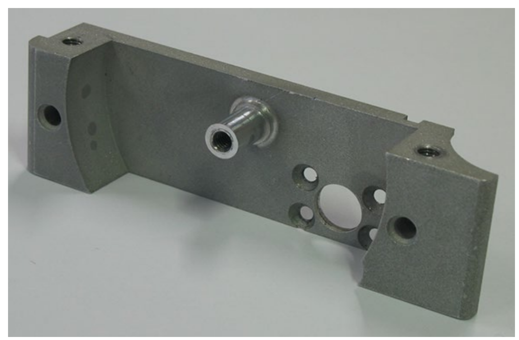
Industrial (automotive) Production

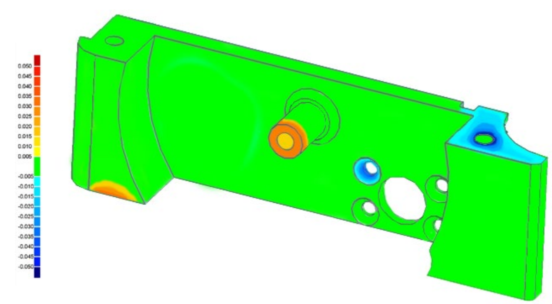
Computer Aided Design (CAD)

Computer Numerical Control (CNC)

Difference CAD - CNC

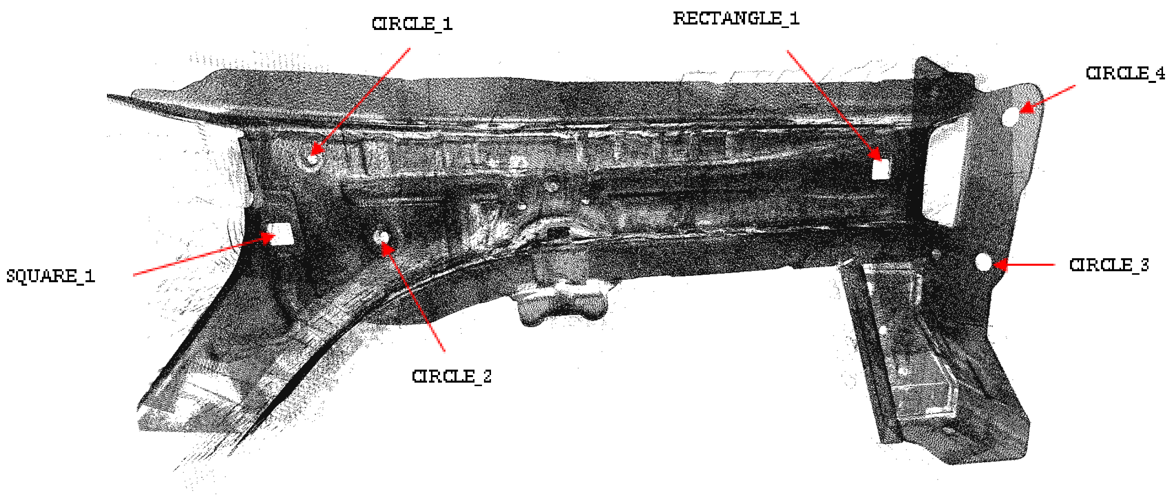






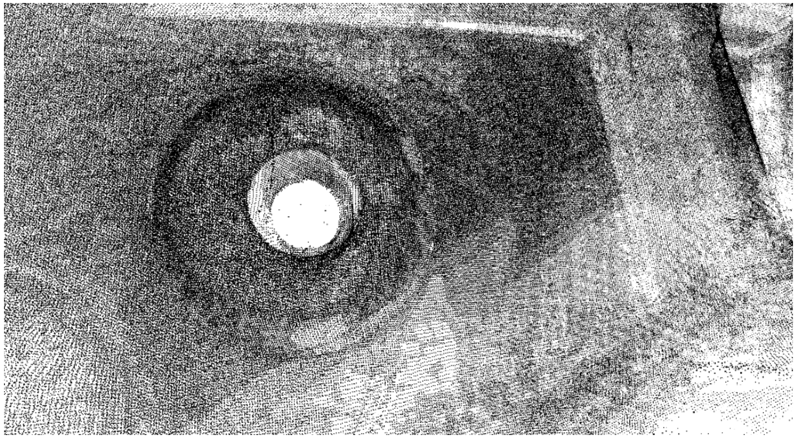
Development

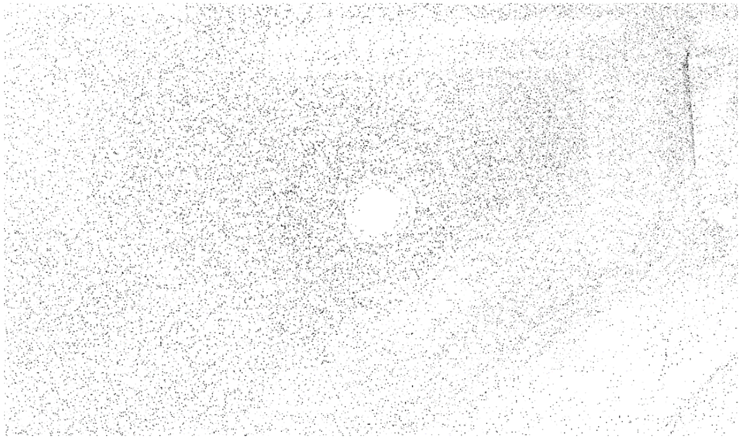
Data



IDENTIFIER	X (MM)	Y (MM)	Z (MM)	DIAMETER(MM)
RECTANGLE_1	-442.0	467.6	348.5	20 - 25
SQUARE_1	100.0	405.0	308.0	20
CIRCLE_1	61.1	415.4	371.0	24.0
CIRCLE_2	9.9	430.5	302.5	14.0
CIRCLE_3	-491.012	536.68	266.12	16.0
CIRCLE_4	-488.873	545.532	381.389	14.5

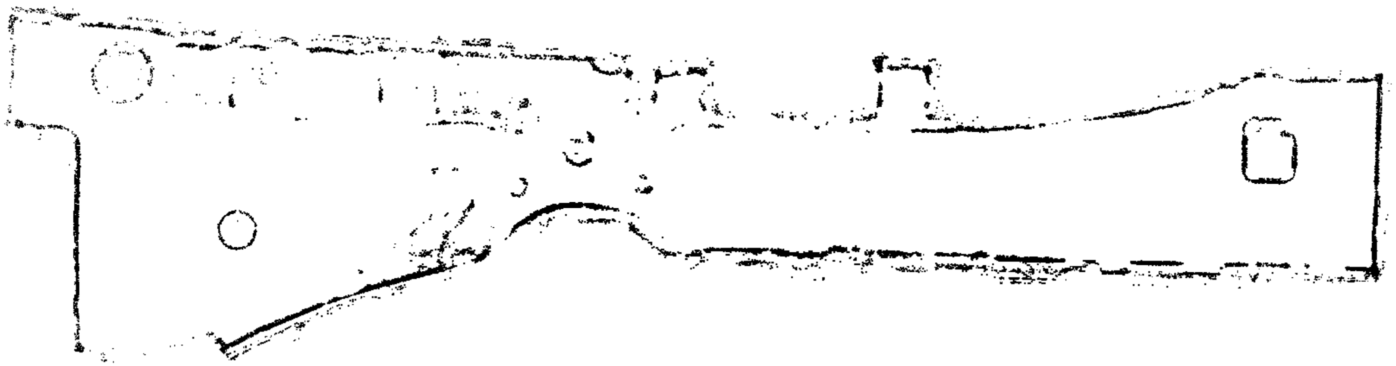
Decimation



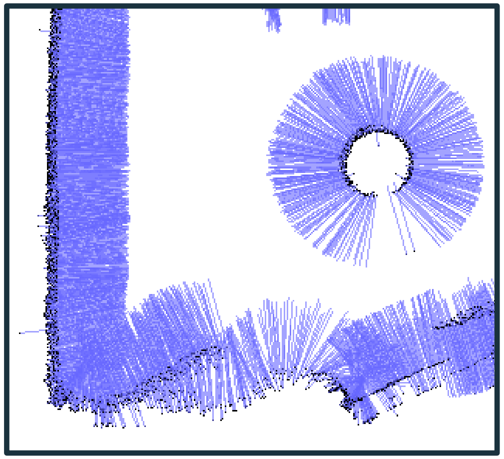


Plane Segmentation

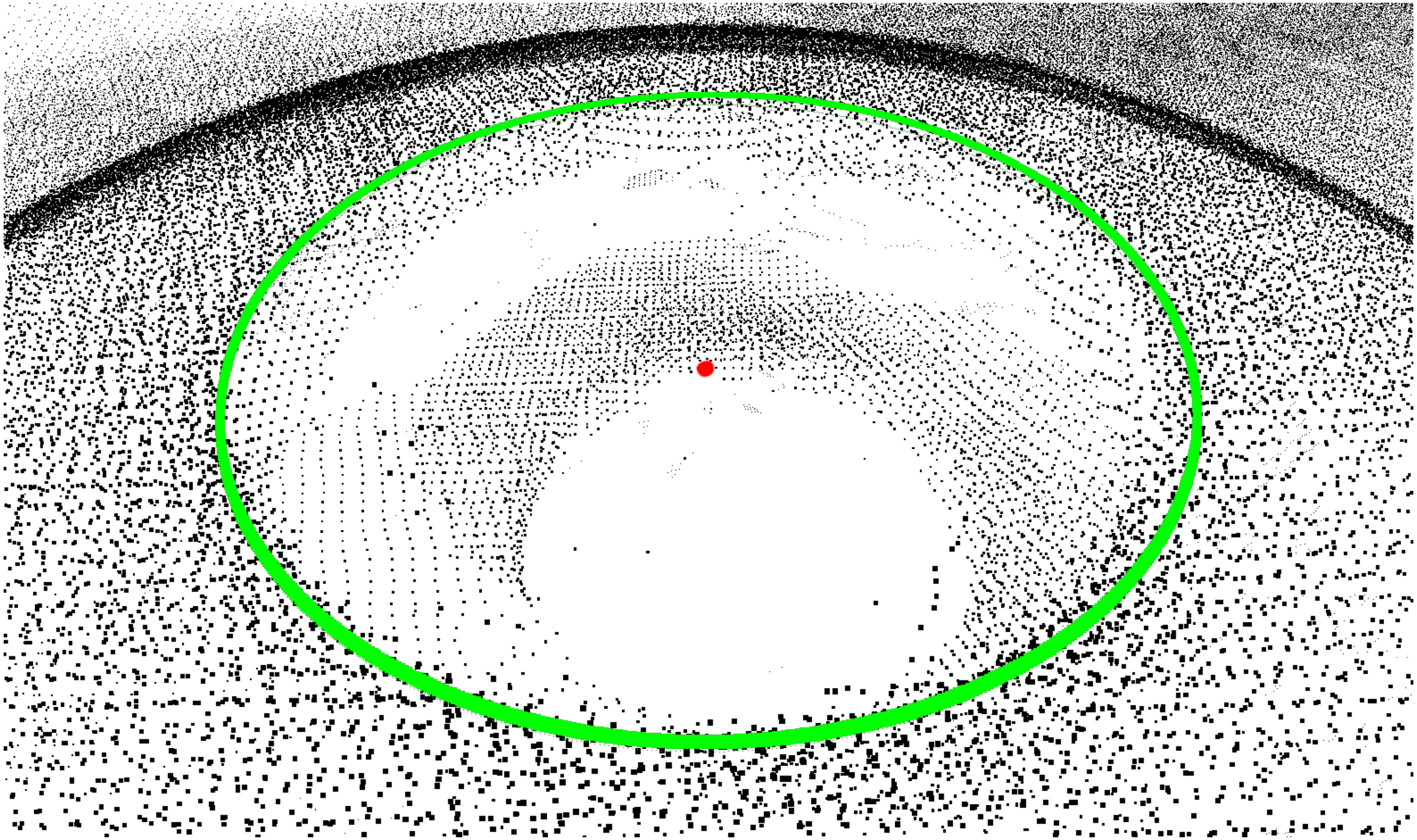
Edge Detection



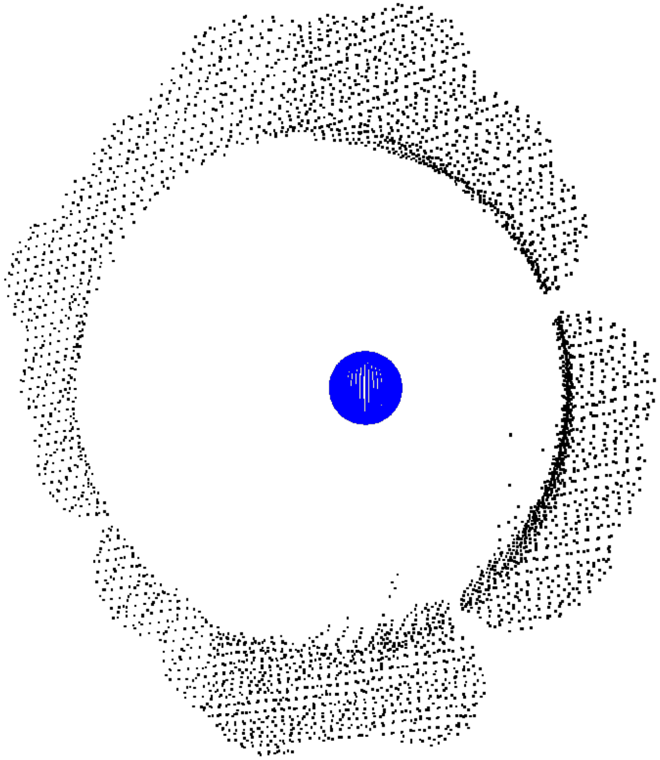
Edge Clustering



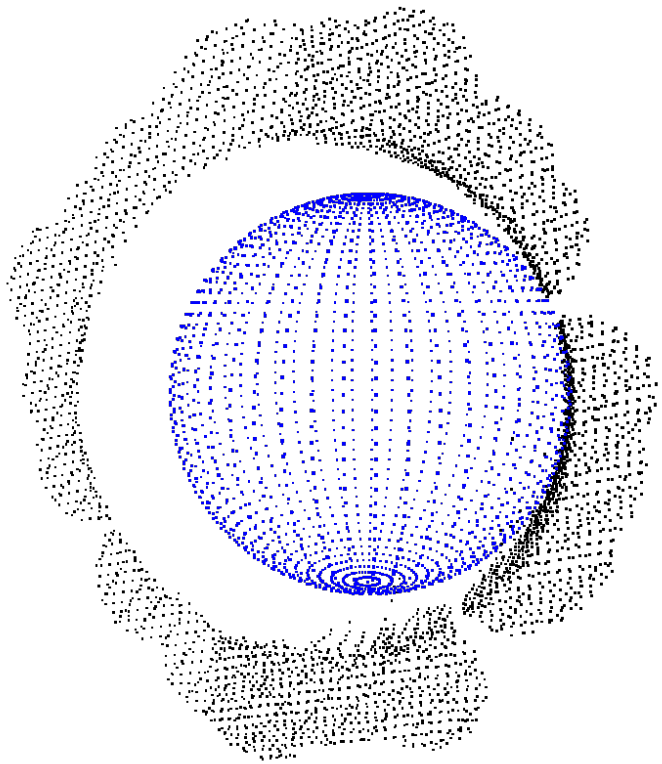
Coarse Detection



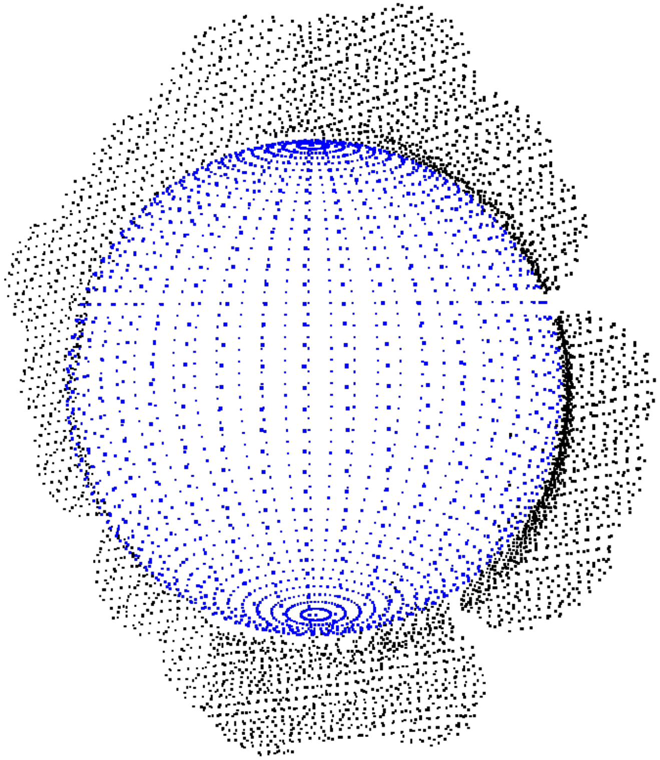
Fine Extraction



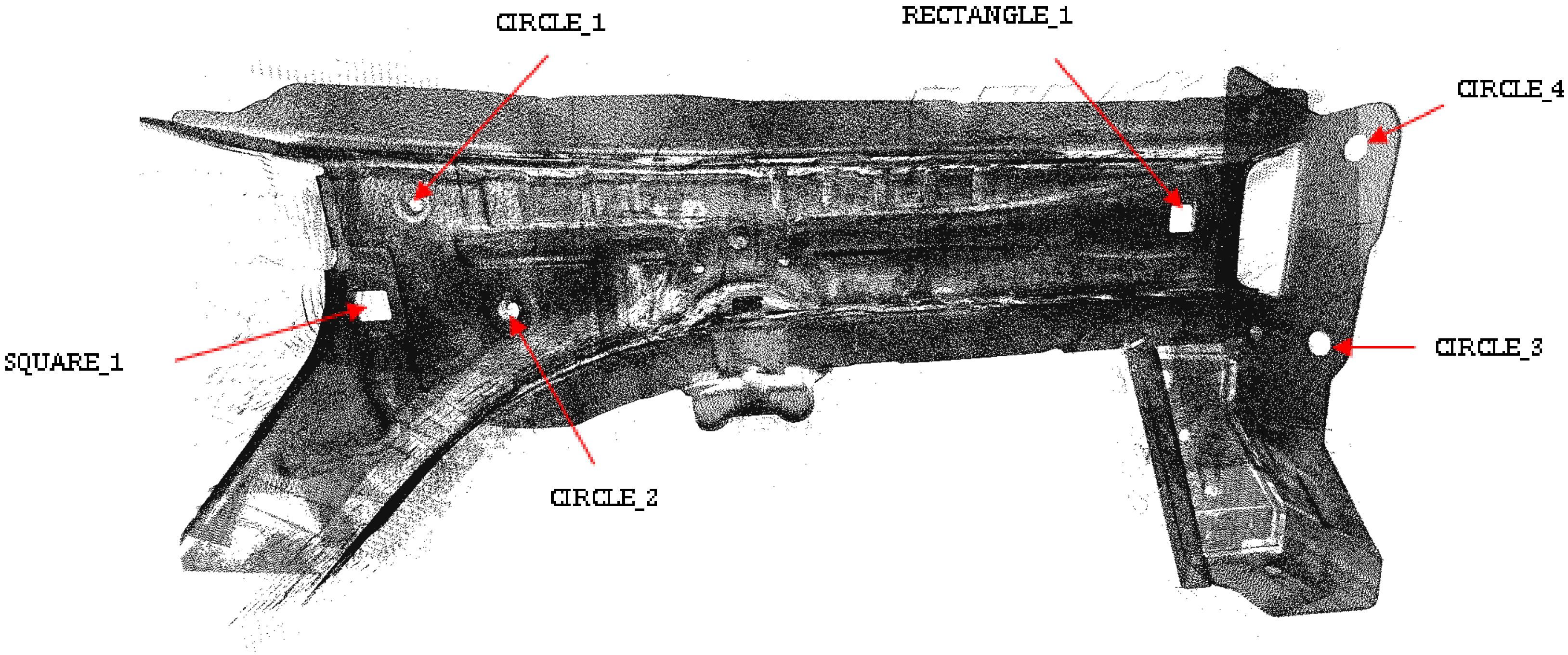
Fine Extraction



Fine Extraction



Results



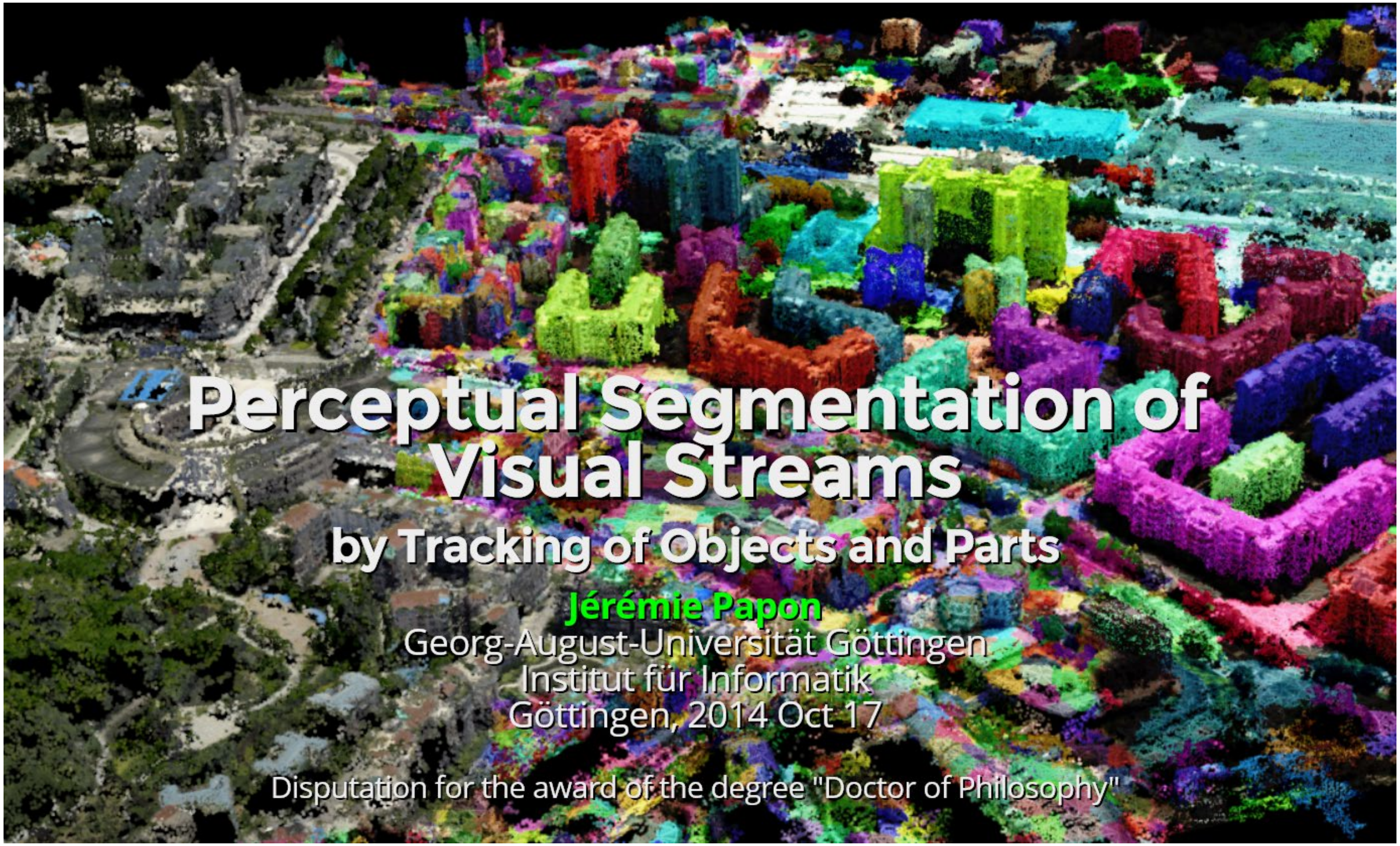
Jeremy Papon, Perceptual Segmentation of Visual Streams

VIDEO PROCESSING 4 POINT CLOUDS (VISUAL STREAMS)

Visual Streams, i.e. Point Cloud VIDEOS!!!...

Supervoxels for Stream Data

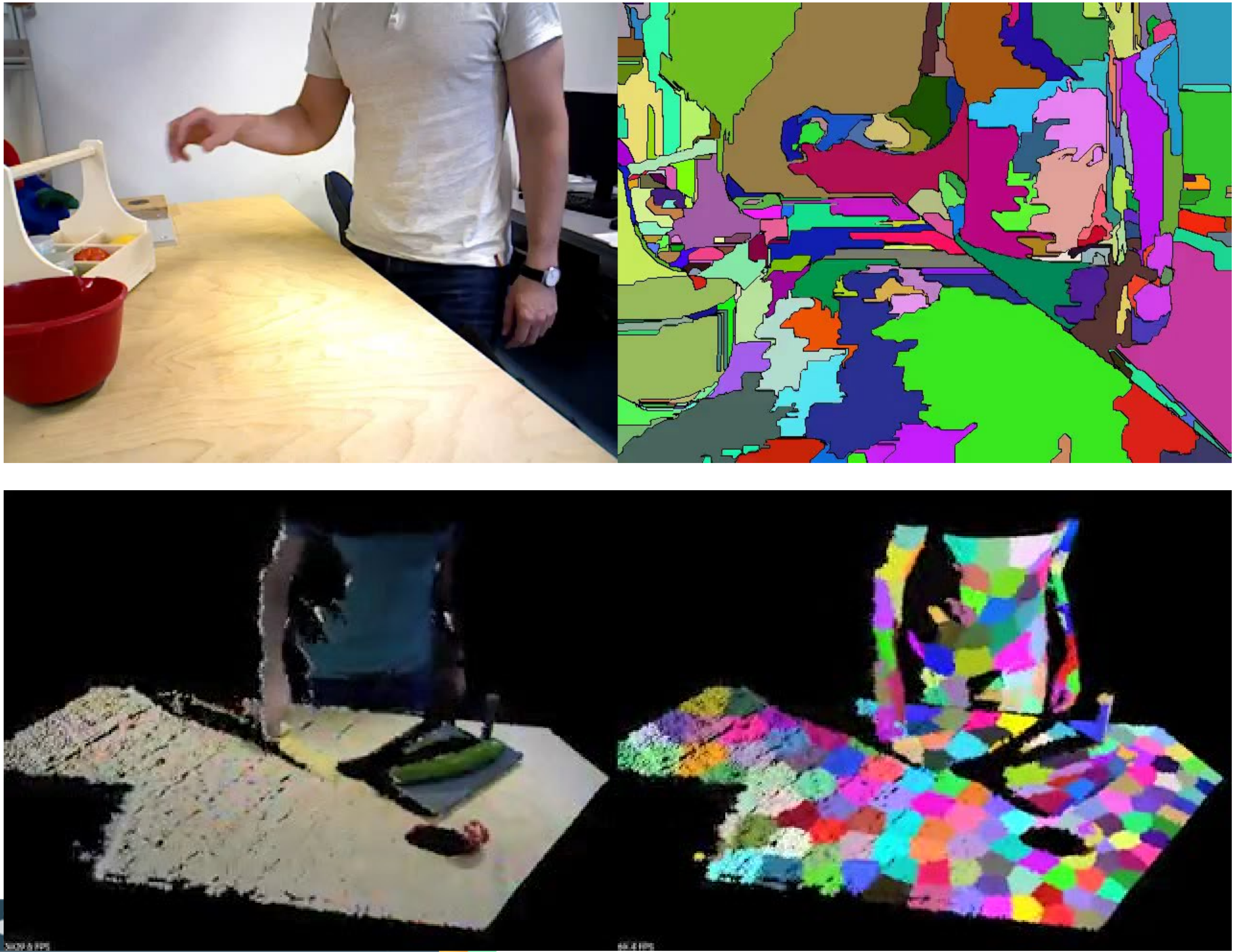
[Papon 2013]



http://jpapon.github.io

Supervoxels for Stream Data

[Papon 2013]

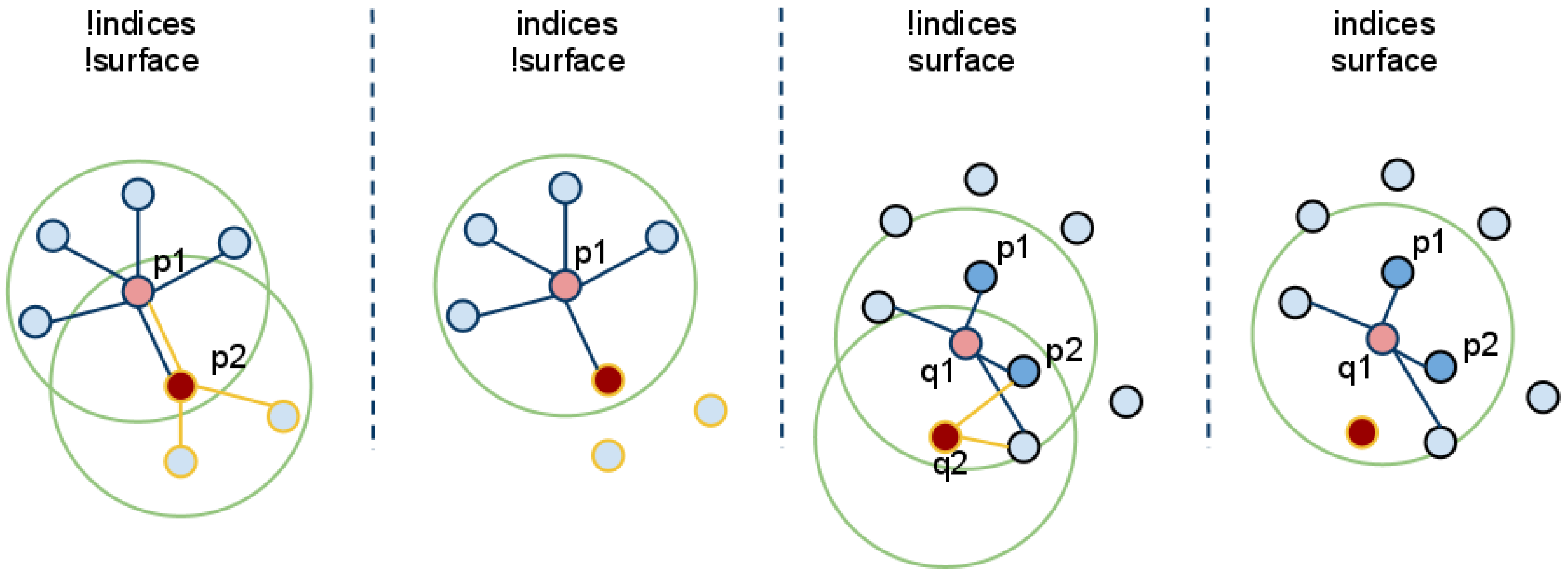


From supervoxels to graphs
in HCI applications

- Generic Video Segmentation on 3D Point Clouds (temporally coherent)
  - Supervoxels are generated using octrees, representing adjacency information on voxels
  - More complex applications with dynamics and occlusions, may require more complex graph structures

Unorganized Point Cloud Processing

Importance of the local neighborhood



(Sensor) Fusion and Graphs
Image Processing on Graphs

Representation and Processing of large/heterogeneous information:

- Often, original data appears unstructured and is based on simple or numerous primitives (pixels, points)

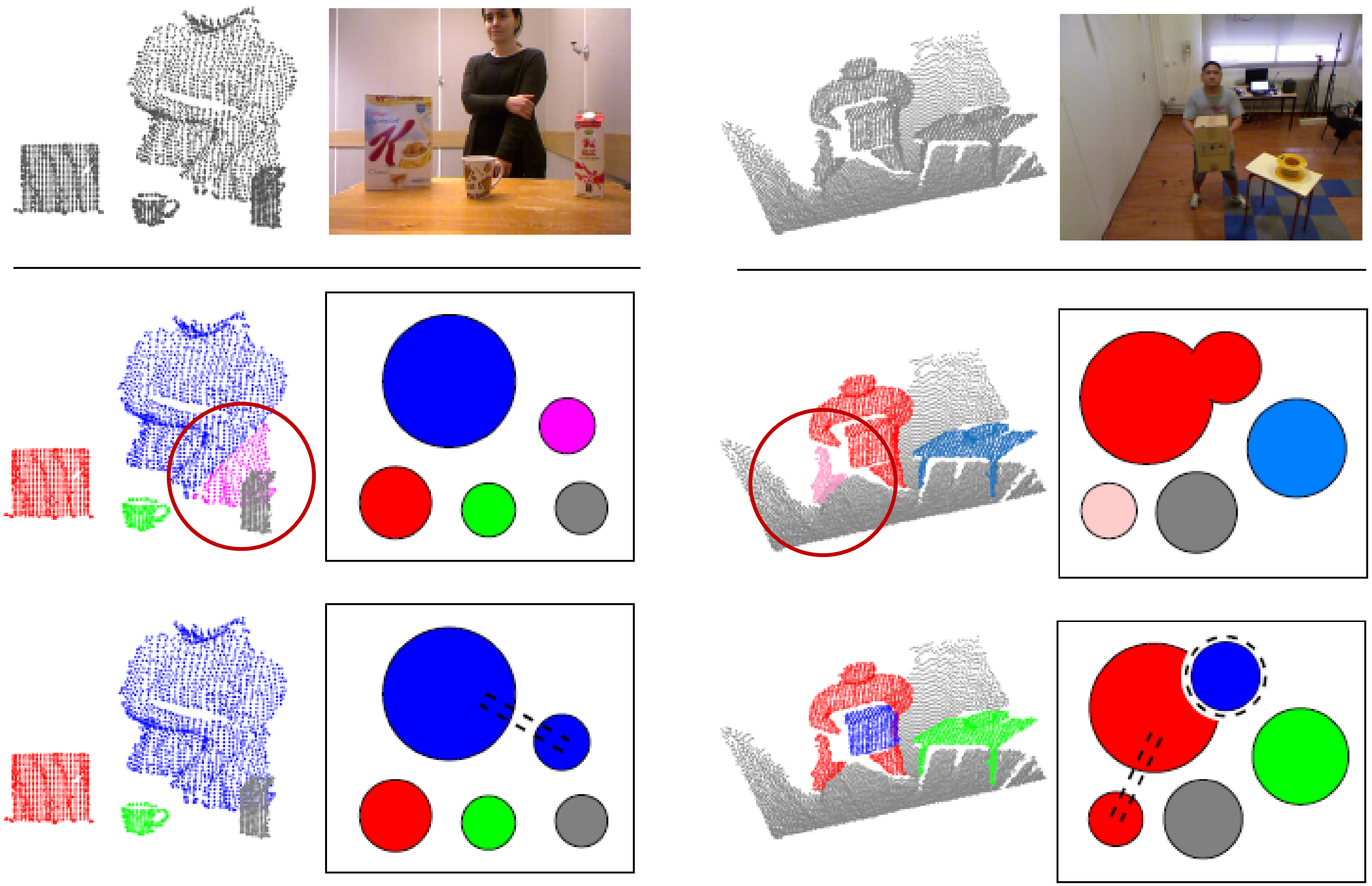
  → Define meaningful primitives, using Graphs or Trees:
  - Nodes with meaningful primitives (possibly heterogeneous)
  - Edges represent relationships… and hierarchies!
  - Data pooling enriches edges and nodes with features/classes
  - Suited for parallel computing → Processing!

- Graph/Tree structures for Big Data applications: genomics, connectomics, *geometric and temporal 3D applications*… inferring data structures for advanced processing on graphs

From supervoxels to graphs

in HCI applications

Scene 1 Scene 2



- Point clouds
- RGB images

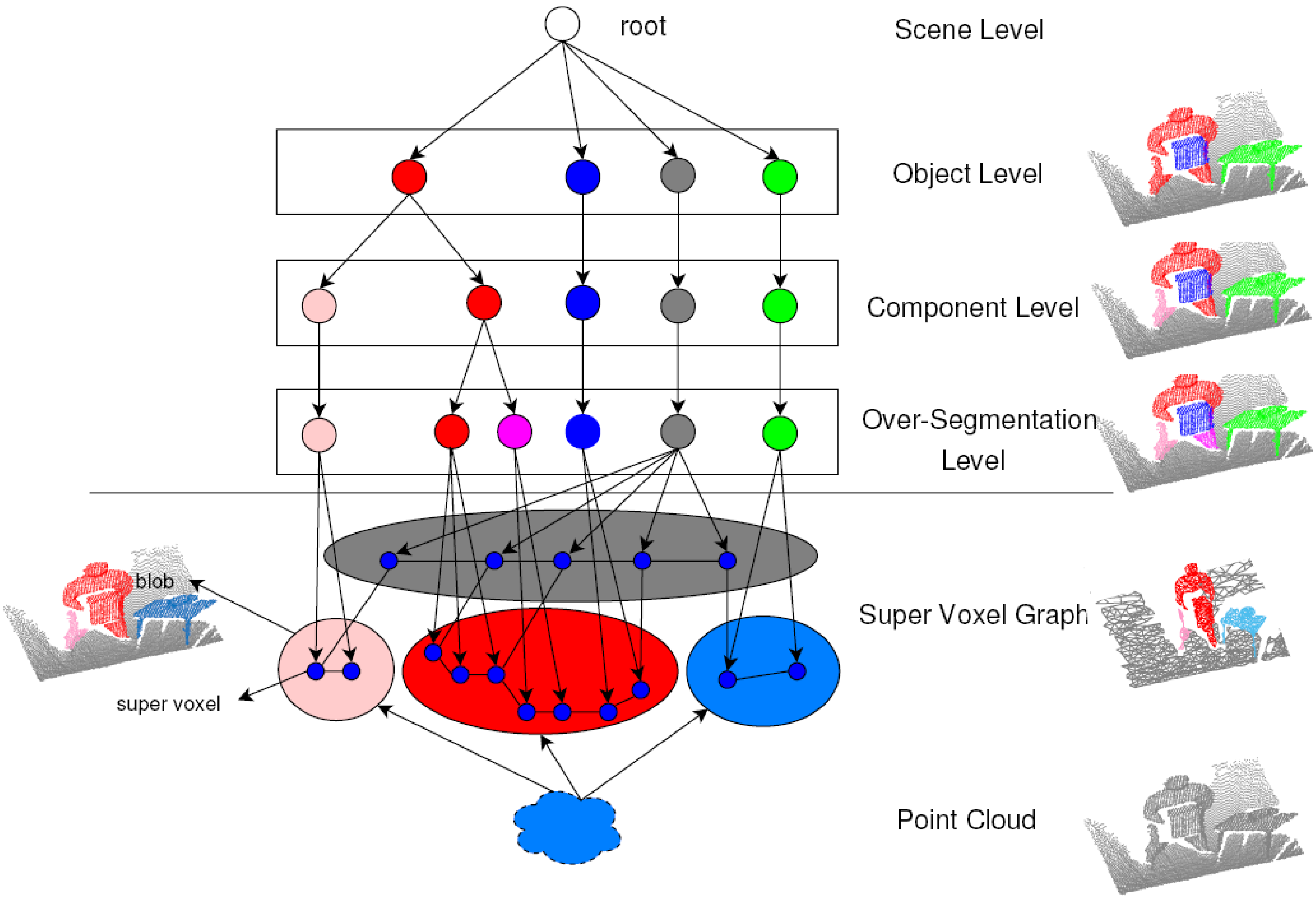
- Segmentation errors
- Sketch maps

Generic segmentation

Spatio-temporal reasoning from higher level features on trees

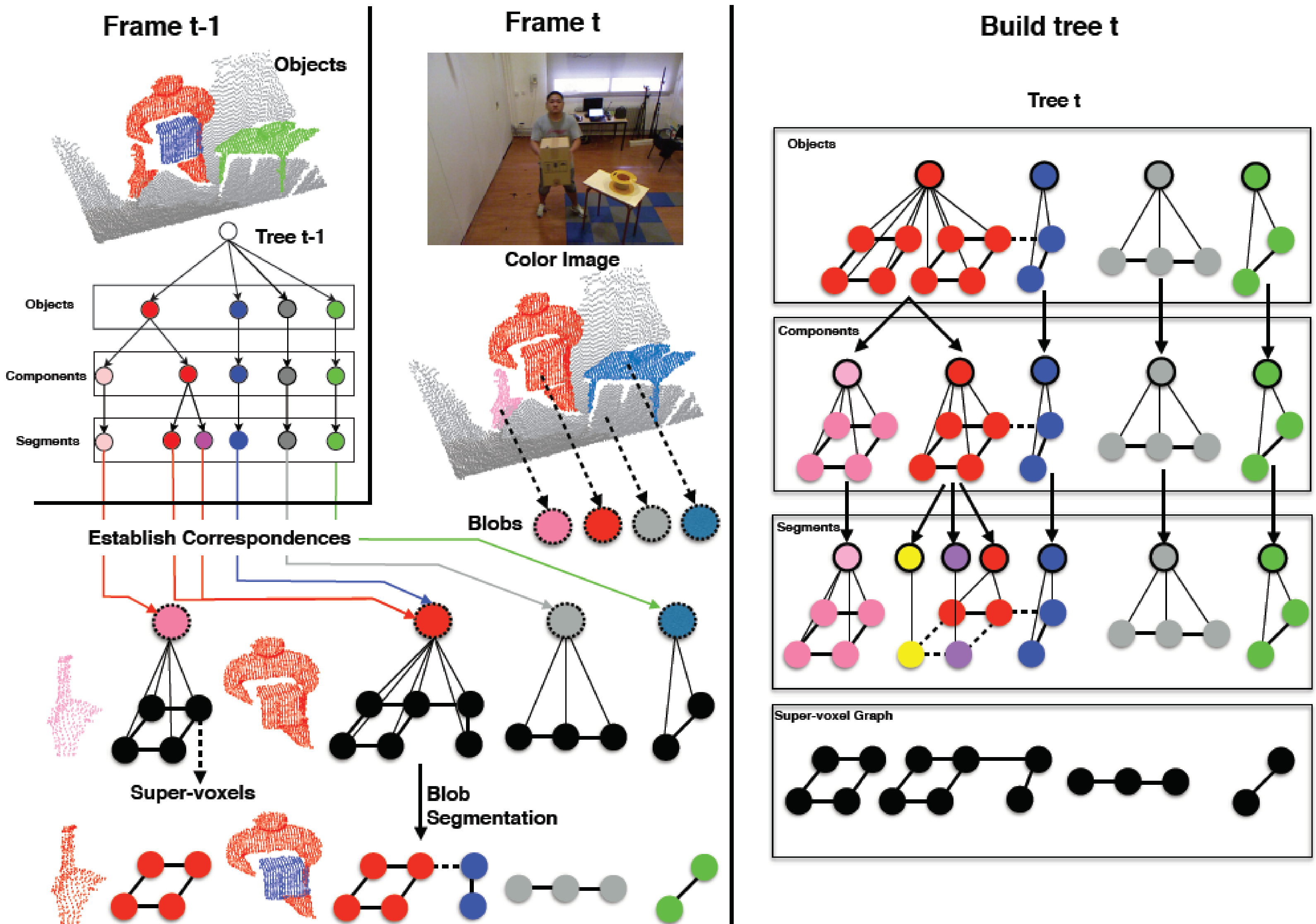
Processing on Trees

[Lin 2018]



Processing on Trees

[Lin 2018]



Processing on Trees
[Lin 2018]

- Generic segmentation approach for 3D point cloud video (stream data)
  - Exploits photometry + geometry
  - Only based on low level features

    (connectivity and compactness)
  - Objects in a single frame represented with a hierarchy

    (tree structure)
  - Tree propagated in time, temporal correspondences

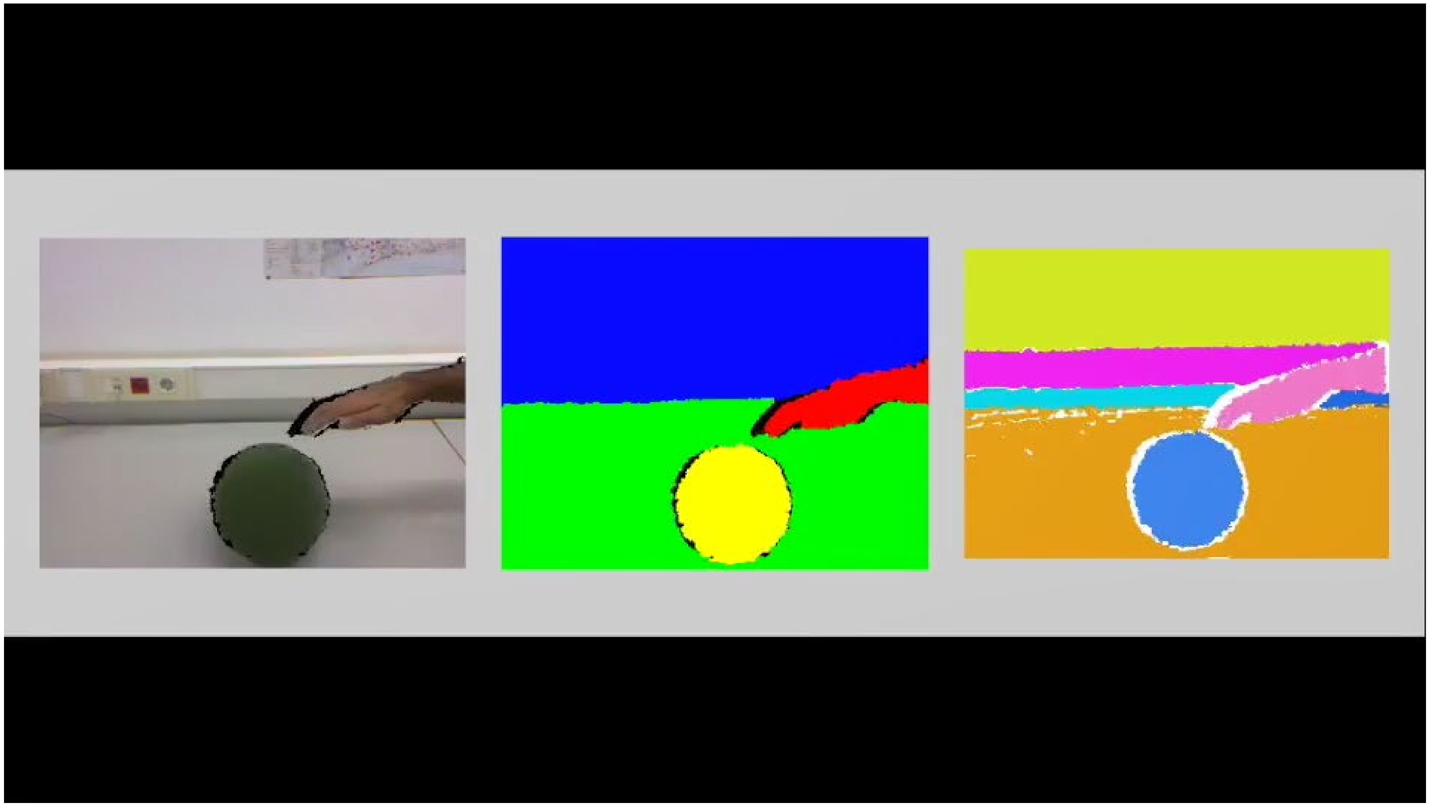
    (at different scales of object-connectivity)
  - Management of splits and merges of objects

    Allows updating segments according to observed evidences

Processing on Trees

Results [Lin 2018]

Original | Tree (Lin 2018) | Surface* (Husain 2015)



Segmentation accuracy

	(Lin 2018)	(Husain 2015)
seq1	99.3	99.5
seq2	82.1	86.0
seq3	77.4	91.8
average	84.8	92.8

* Maintains a quadratic surface model to represent the object segments

Processing on Trees

Results [Lin 2018]

Generic Tree (Lin 2018) | Object proposals* (Fu 2017)

* Objects from a pool of object proposals (graph optimization, objectness, motion…)
  – computationally expensive
  – can not handle varying number of objs.

Deep Learning with sets and Point Clouds

PointNet++:

Neural network that directly consumes point clouds

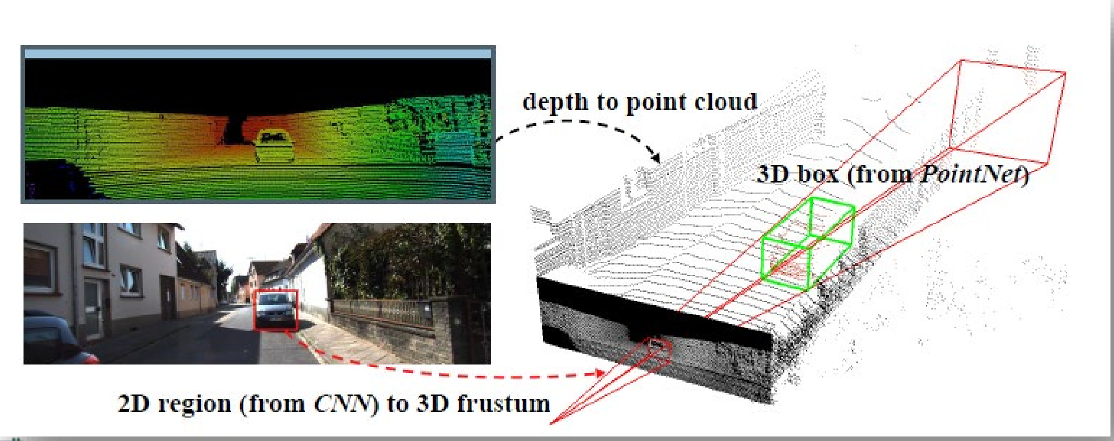
- respects the permutation invariance of points
- each point is processed identically and independently, represented by just its three coordinates (x, y, z)
- additional dimensions added: normals and local/global features
- provides a unified architecture for applications: object classification, part segmentation, scene semantic parsing

[Qi 2016] [Qi 2017] stanford.edu/~rqi/pointnet

Frustum PointNet

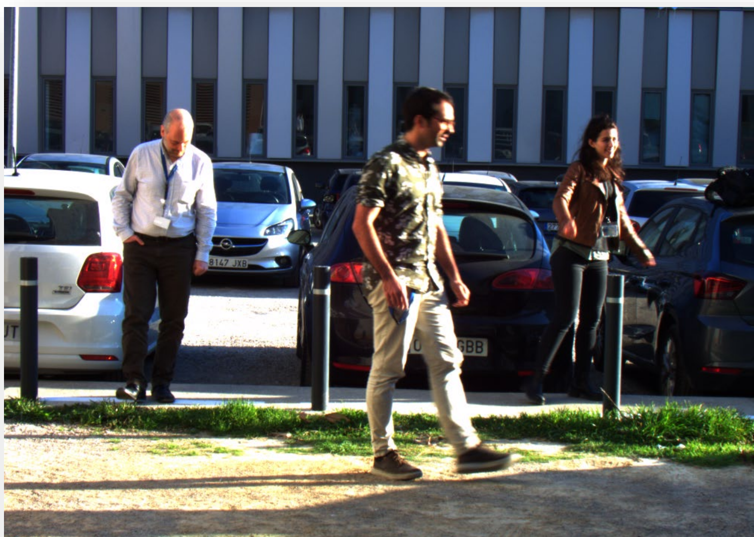
- Fuses detections in RGB and PointClouds

[Qi 2018]

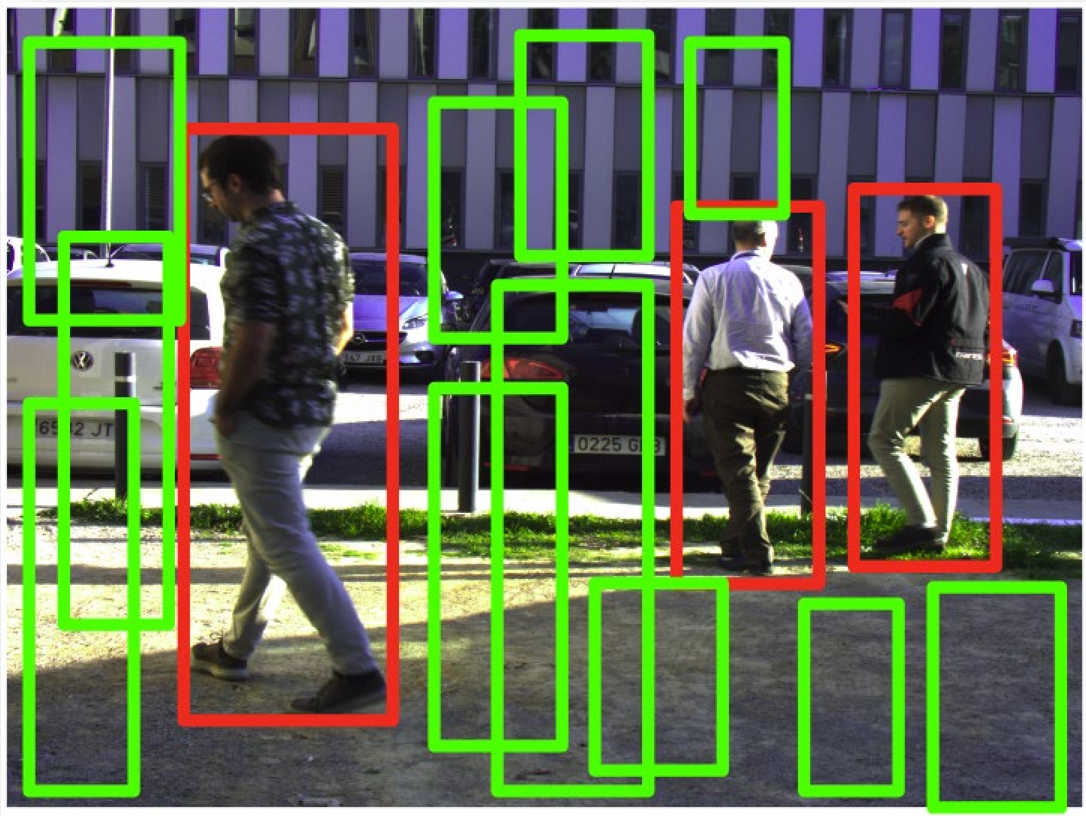


O. Lorente, Pedestrian Detection in 3D Point Clouds using Deep Neural Networks

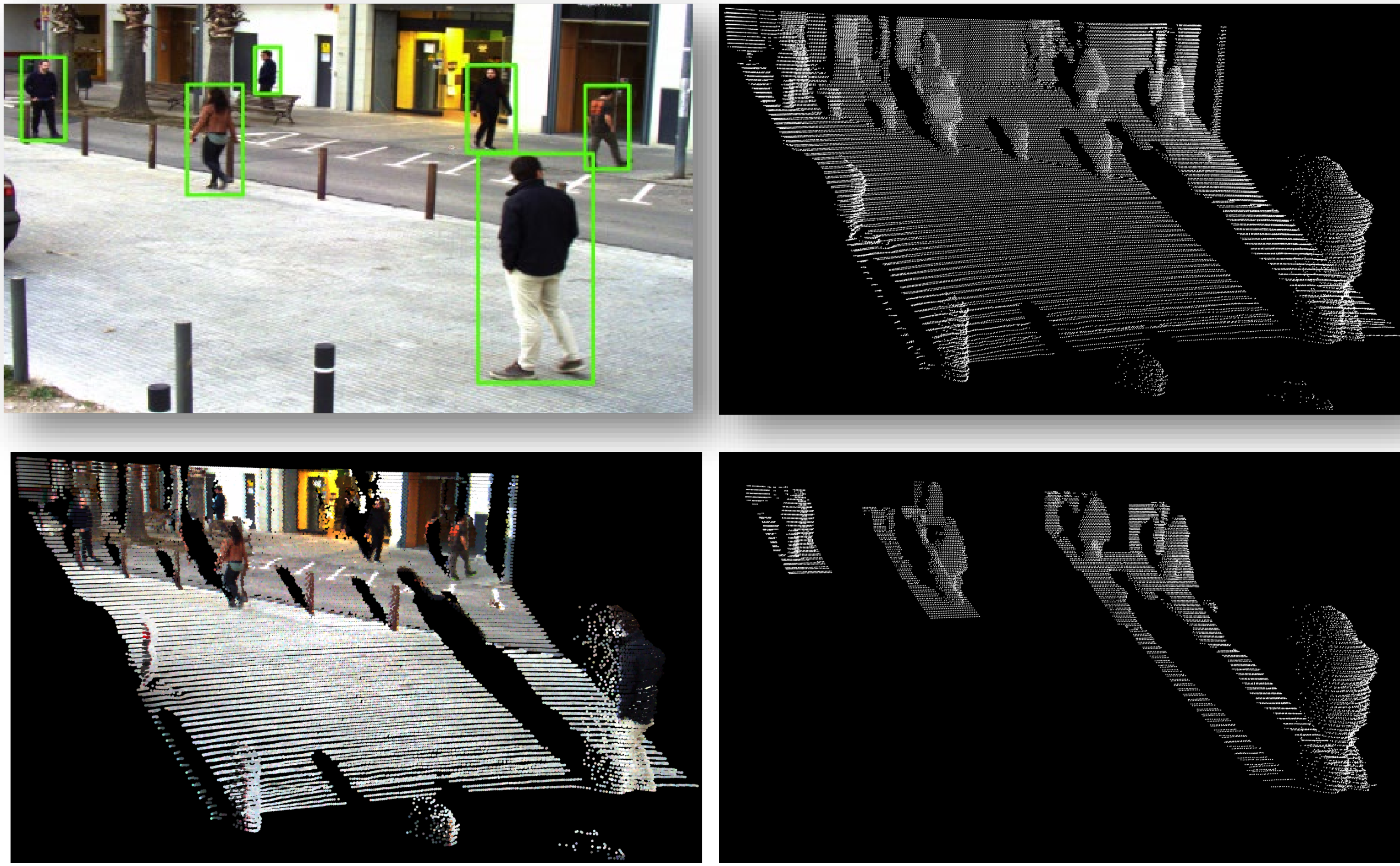
Degree Thesis ETSETB/UPC 2019



Pedestrian / Non-pedestrian



Backprojection to Frustum



Ò. Lorente Corominas, Pedestrian detection in point clouds, DegTh TelecomBCN UPC 2020

Improved architectures

C. R. Qi et al, "Frustum PointNets for 3D Object Detection From RGB-D Data," CVPR 2018

- Leverages mature 2D detectors and 3D deep learning (PointNet++) for object localization for even small objects (3D bounding boxes)

S. Shi et al, "PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud," CVPR 2019

- Stage-1: bottom-up 3D proposals (foreground segmentation)
  Stage-2: local spatial features from pooled points of 3D proposals combined with global semantic features for box refinement

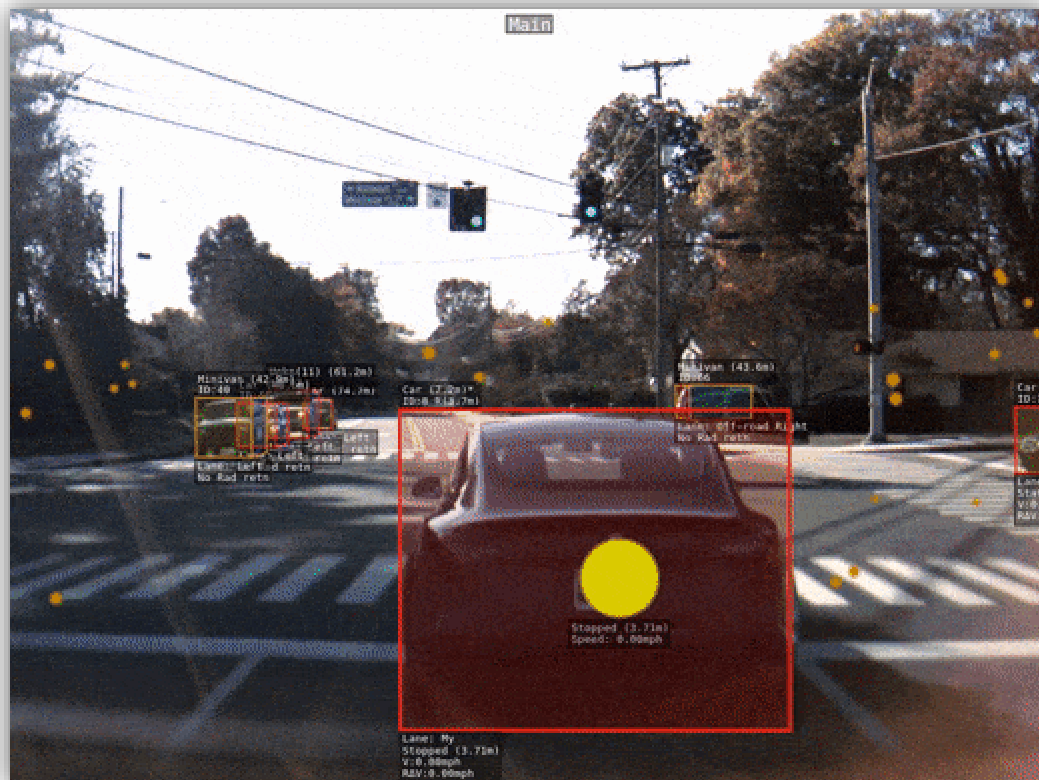
Shaoshuai Shi et al, "From Points to Parts: 3D Object Detection from Point Cloud with Part-aware and Part-aggregation Network, " TPAMI 2020

- Stage 1: part-aware predicts 3D proposals and intra-object part locations, grouped by RoI-aware point cloud pooling (geometry).
- Stage 2: part-aggregation: re-scores and refines box location by exploring the spatial relationships of intra-object parts.

…and what about RADAR?

Applications with 4D Imaging Radar

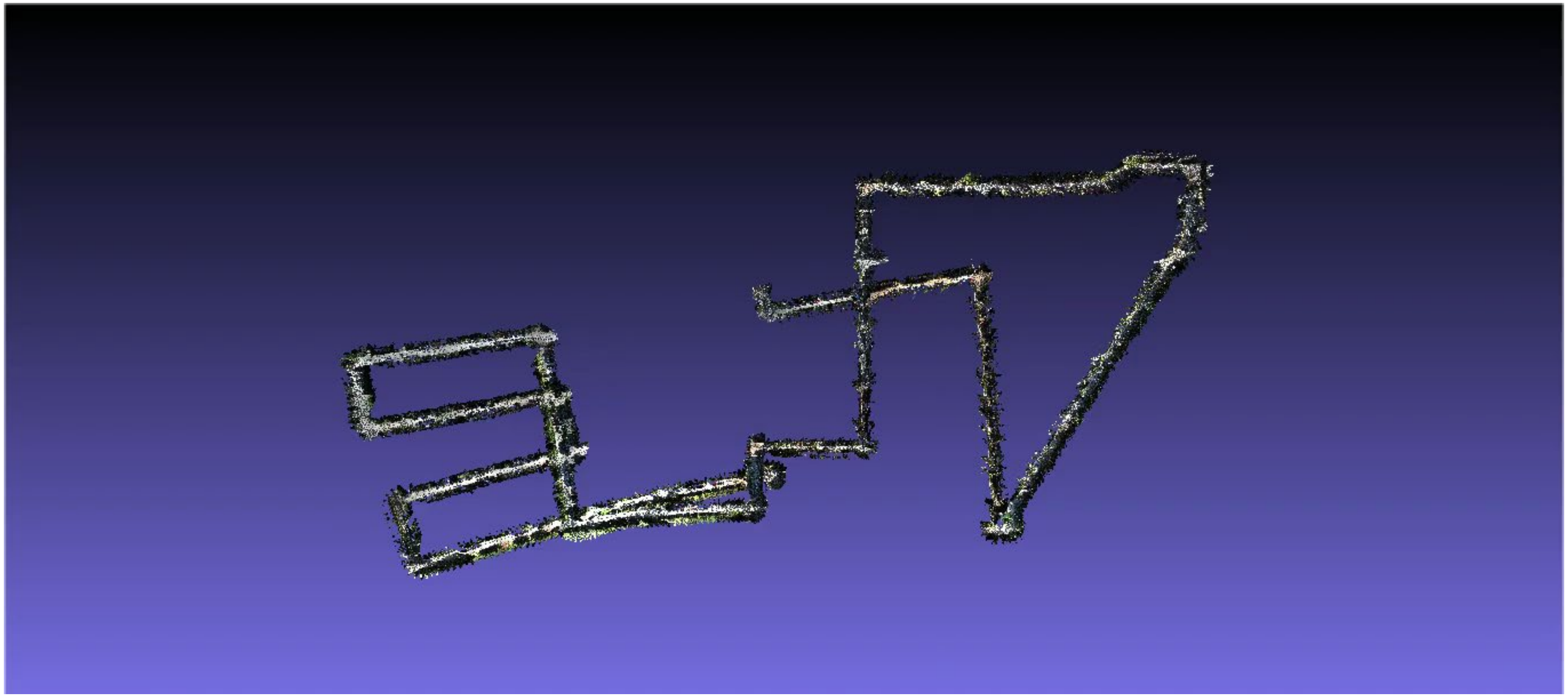
Have seen this...



(source)

Discussion

- LiDAR is already much sparser than video…



[Video](#) [PLY](#)

Discussion

- LiDAR is already much sparser than video…

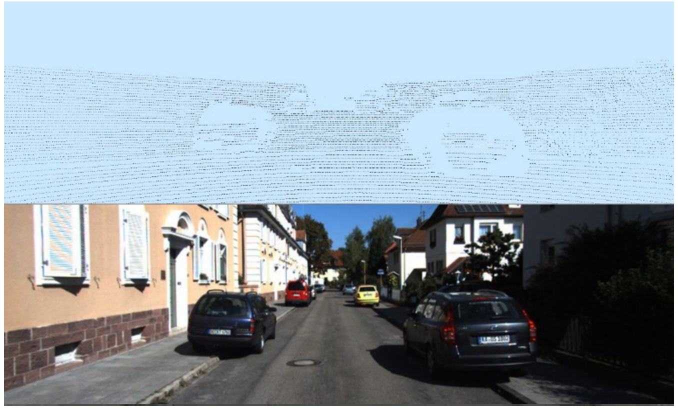
…

Discussion

- LiDAR is already much sparser than video…
  …and we might need to interpolate

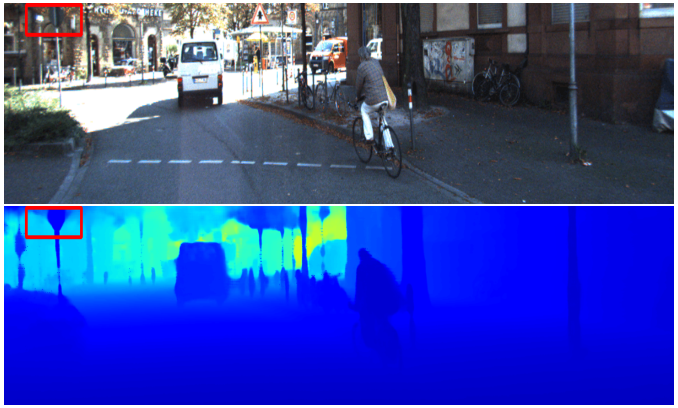
  Estimate *object range* (constant) and *depth/motion profiles* from radar detections. Advantage: helps localizing and improving video detections

(a) Morphological Depth Interpolation



From sparse LiDAR points (top) to depth profiles (bottom) on Kitti dataset
I. Caminal et al (2018), "*Slam-Based 3D Outdoor Reconstructions from Lidar Data*," IC3D
10.1109/IC3D.2018.8657869

(b) DL architecture: 2 branches + GCL



Dense from sparse Depth maps with Image Guided Completion on Kitti dataset
M. Hu et al (2021), "PENet: Towards Precise and Efficient Image Guided Depth Completion," ICRA
10.1109/ICRA48506.2021.9561035

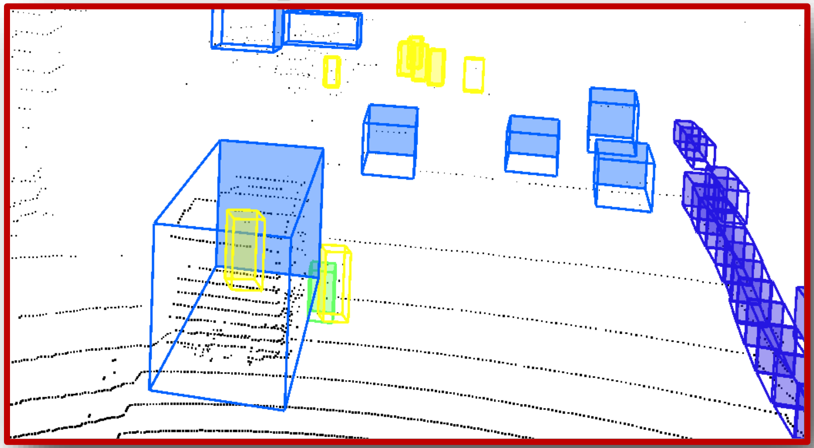
Discussion

…but RADAR is even more sparse!

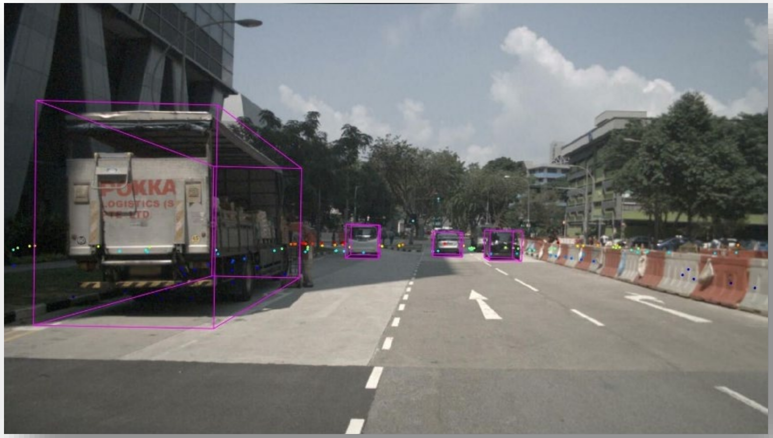
In ego_pose (IMU) 3D coordinate system

& LiDAR (pointcloud)

Projected to 2D frame



Scene name = 'scene-0061'
Sample_token = 'ca9a282c9e77460f8360f564131a8af5

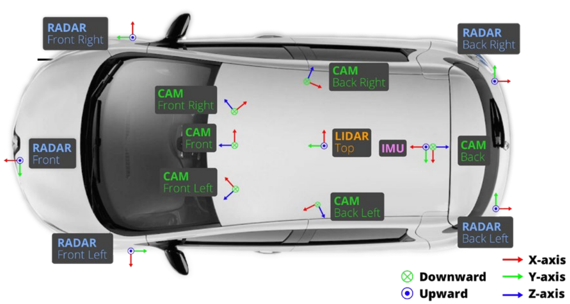


Discussion: datasets

nuScenes

nutonomy

- Outdoor (**CAMS+RAD+LIDAR**)
- 1,000 scenes!
- Ground Truth:
  - Trajectory
- Metrics:
  - RPE



1x spinning LIDAR (Velodyne HDL32E):
20Hz, 32ch, FoV 360°x(+10°:-30°), range/acc 70m/±2cm

5x long range RADAR sensor (Continental ARS 408-21):
13Hz, 77GHz, FMCW dist&vel, rang.250m, vel.acc ±0.1 kmh

6x camera (Basler acA1600-60gc):
12Hz, 1600x900 Bayer8 1bpp, JPEG

1x IMU & GPS (Advanced Navigation Spatial):
Accuracies: pos.20mm, head.0.2° w/GNSS, Roll&Pitch 0.1°

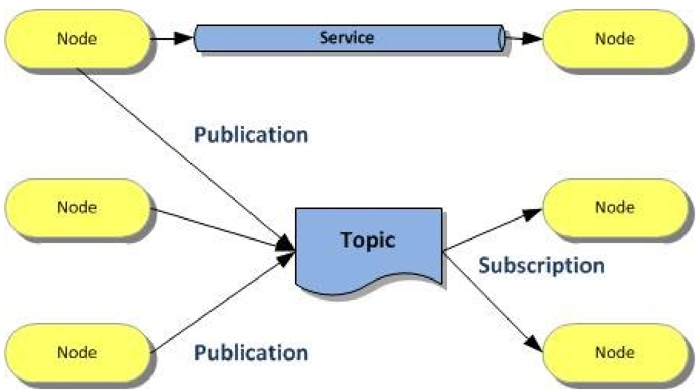
Website, paper: CVPR 2020, github: nuscenes-devkit

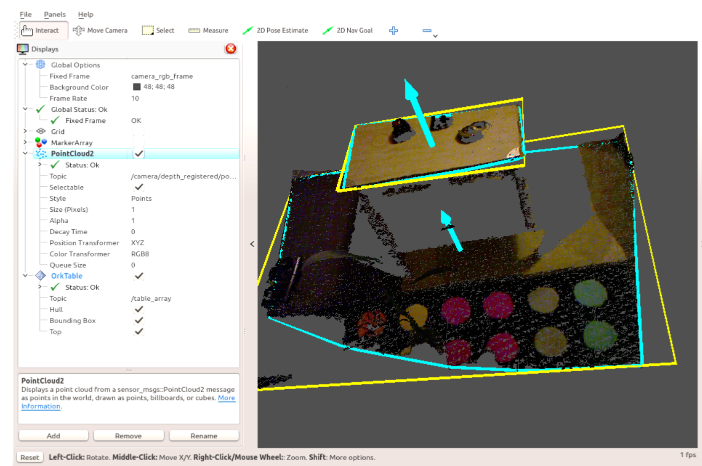
explore dataset

A note on Distributed Processing

Robot Operating System (ROS)

- OS vs Software Development Framework
  - Package Manager, Package Repositories, Metabuilder (Python/C++)
- Distributed Message, Services and Config (Nodes, Nodelets and Master)
  - Scientific packages (robotics), industrial impact
  - Centralized and structured logging (rosout)
  - Generic recording/save system (rosbag)
  - Time synchronization and emulation (clock)
  - Generic Visualization (rviz)
  - Adapted to computing services (srun vglrun rosrun + tmux)





- Not only for Robotic Apps/Demos
- But for **replicable research, with distributed processing**

What to do with the results?

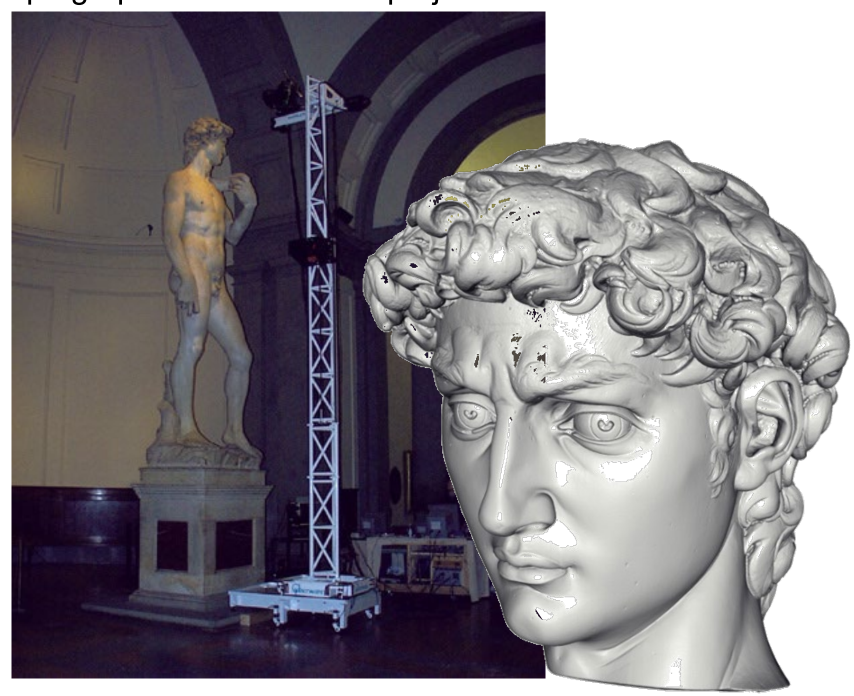
MESHING

Motivation for meshing

In many domains, scanners are used to obtain virtual representations of 3D shapes

http://www.jhu.edu/digitalhammurabi/

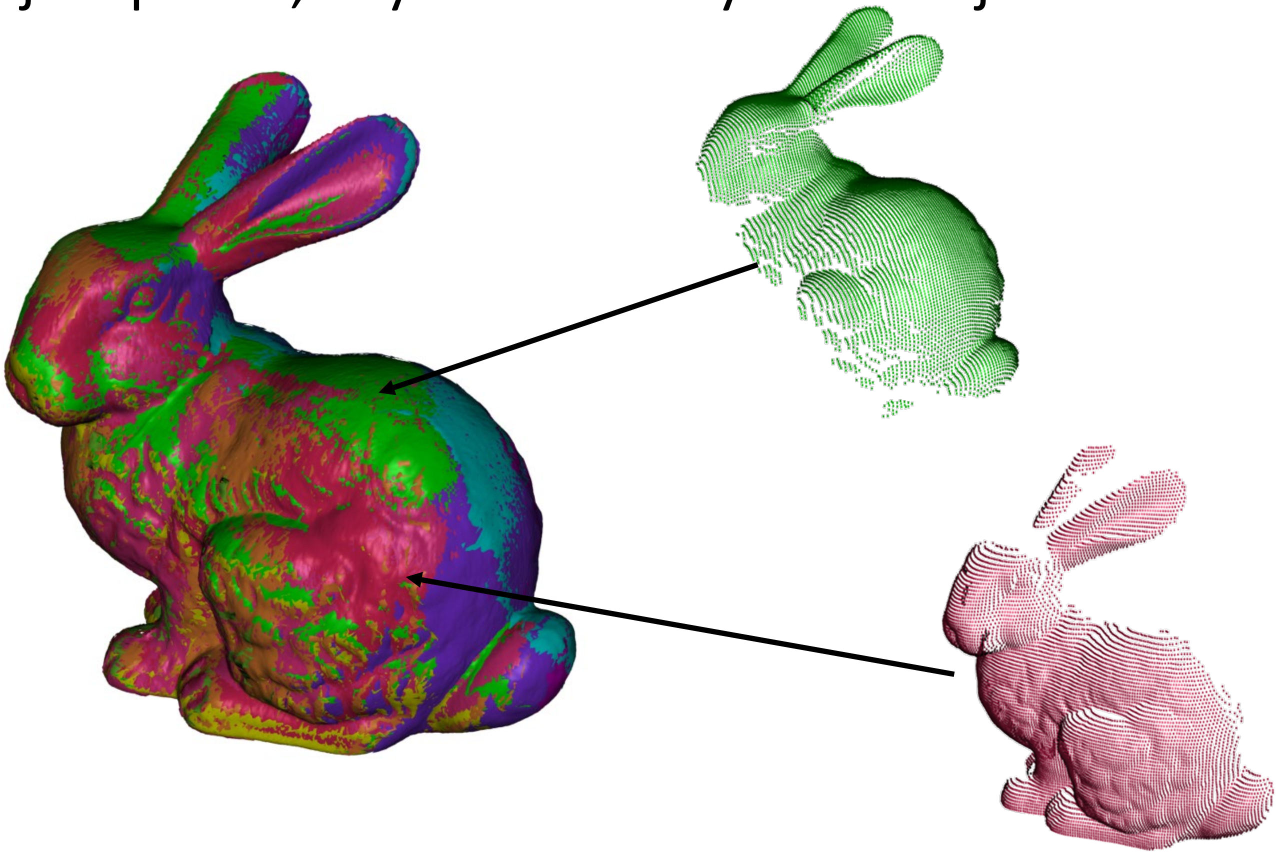
http://graphics.stanford.edu/projects/mich/



Scanner results

Scanning often gives only local connectivity…

…or even just points, any connectivity at all → just a PointCloud!

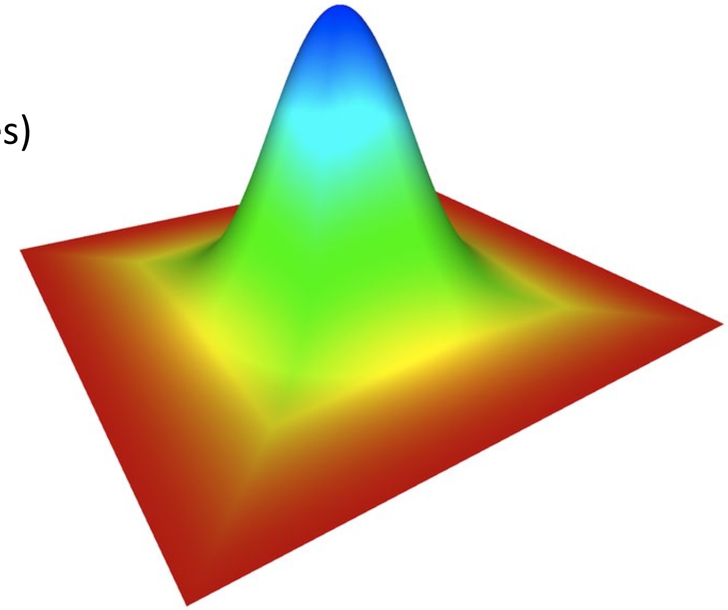


but yet... There is some motivation for meshing

We want a 3D Mesh for:

- Parameterization
- Computational Analysis
- Rapid Prototyping
- Rendering
- Collision Detection

Meshing



Mesh

Defines an ordering of the surface points (vertices) for interpolating a continuous surface in intermediate positions

Strategy

Greedy contour propagation under a set of rules for topological correctness

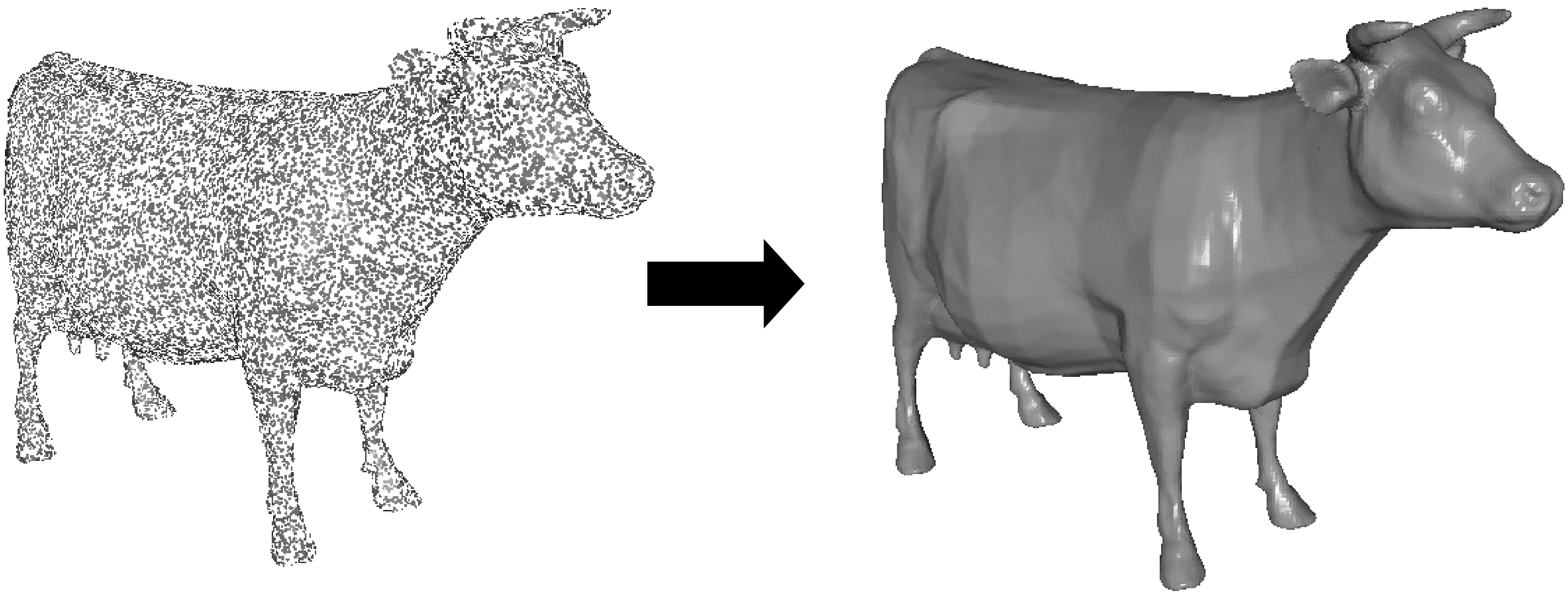
Implementation

Efficient 3D spatial queries of neighbor points via *kd-tree* applied iteratively for processing disconnected sets of points

Surface Reconstruction

M. Kazhdan *Johns Hopkins Univ. Poisson Reconstruction Eurographics 2006*

Generate a mesh from a set of surface samples



Surface Reconstruction

Generate a mesh from a set of surface samples

- Three general approaches:
  1. Computational Geometry

     Boissonnat, 1984
     Edelsbrunner, 1984
     Amenta *et al*., 1998
     Dey *et al*., 2003

  2. Surface Fitting

     Terzopoulos *et al.*, 1991
     Chen *et al.*, 1995

  3. Implicit Function Fitting

     Hoppe *et al.*, 1992
     Curless *et al.*, 1996
     Whitaker, 1998
     Carr *et al.*, 2001
     Davis *et al.*, 2002
     Ohtake *et al.*, 2004
     Turk *et al.*, 2004
     Shen *et al.*, 2004
     Kazhdan, 2005

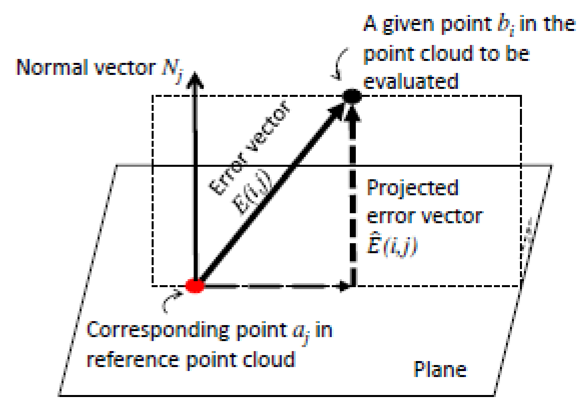
Meshing benchmark

- RMS *Hausdorff* distance between two non-empty subsets *X* and *Y* of a metric space (*M, d*), with *d* the Euclidean distance in 3D space:

$$d_H(X,Y) = \sqrt{\frac{1}{N}\sum_{x\in X} \| \inf_{y\in Y} d(x,y)\|^2}$$

- Computation time
- Memory footprint
- Point cloud distortion metrics
  - Point-to-point
  - Point-to-plane (symmetric)

from Tian *et al*, 2017



Conclusions

- Computer Vision
  - Already a commodity?
  - Increasing complexity:

    image → video
    pointclouds → stream data
  - Multi-sensor fusion (big data)

- Scene understanding
  - Key technology for “Smart-X” (AI)
  - Multi-sensor input
  - Parallel/distributed processing
  - Processing on graphs/trees

References (processing)

[Friedman 1977] Friedman et al, **An algorithm for finding best matches in logarithmic expected time**, ACM Trans. on Math. Software 3(3), 1977

[Fu 2017] H. Fu et al, **Object-based Multiple Foreground Segmentation in RGBD Video**, IEEE Trans. IP 26 (3), 2017. DOI: 10.1109/TIP.2017.2651369

[Husain 2015] F. Husain et al, **Consistent Depth Video Segmentation Using Adaptive Surface Models**, IEEE Trans. Cybern 45 (2), 2015, DOI: 10.1109/TCYB.2014.2324815

[Izadi et al 2011] S. Izadi et al, KinectFusion: **Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera**, ACM Symposium on User Interface Software and Technology, October 2011, DOI: 10.1109/CVPR.2013.264

[Lin 2018] X. Lin et al, **Temporally Coherent 3D Point Cloud Video Segmentation in Generic Scenes**, IEEE Trans. IP 27 (6), June 2018, DOI: 10.1109/TIP.2018.2811541

[Papon 2013] J. Papon et al, **Voxel Cloud Connectivity Segmentation - Supervoxels for Point Clouds**, CVPR 2013

[Rusu 2009] R. B. Rusu, **Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments**, TUM, PhD 2009

[Zhou 2009] Q.-Y. Zhou et al, **Open3D: A Modern Library for 3D Data Processing**, arXiv:1801.09847 2019

References (processing, continued)

[Qi 2016] C.R. Qi et al, **PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation**, *arXiv:1612.00593, 2016*

[Qi 2017] C.R. Qi et al, **PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space**, in Adv. in Neural Information Proc. Sys. 30, 2017, *arXiv:1706.02413*

[Qi 2018] C.R. Qi et al, **Frustum PointNets for 3D Object Detection From RGB-D Data**, CVPR 2018, pp. 918–927 [online]

[Tian 2017] D. Tian et al, **Geometric distortion metrics for point cloud compression**, ICIP 2017, pp. 3460–3464 DOI: 10.1109/ICIP.2017.8296925.