



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
- List (34)
Recording!

Session 10: 3D processing & applications

- Pointcloud processing:
 - PCL (2011) & Open 3D (2018)
 - Organized/Unorganized processing
 - Applications

3D data representation (recall)

- 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...

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?

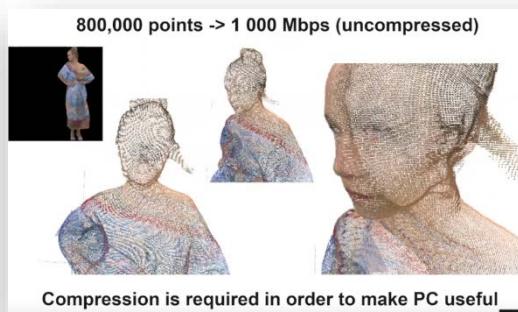
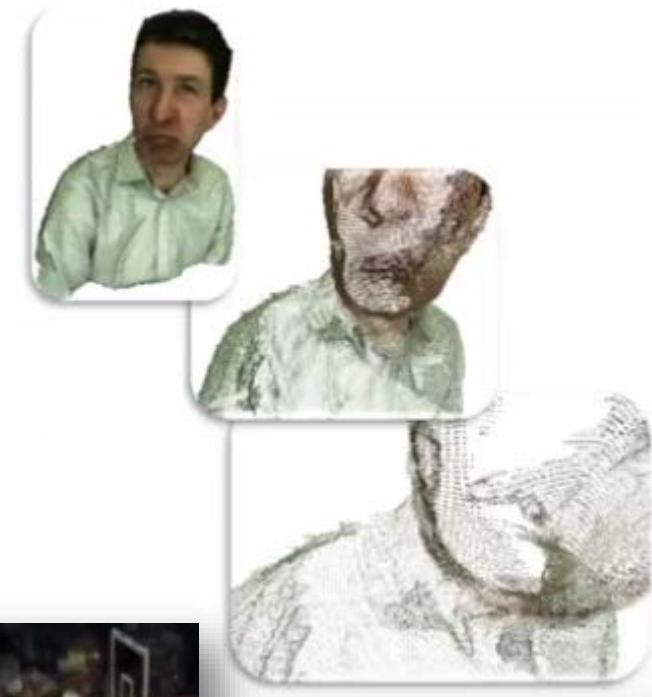


First problem: data management & transfer!

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

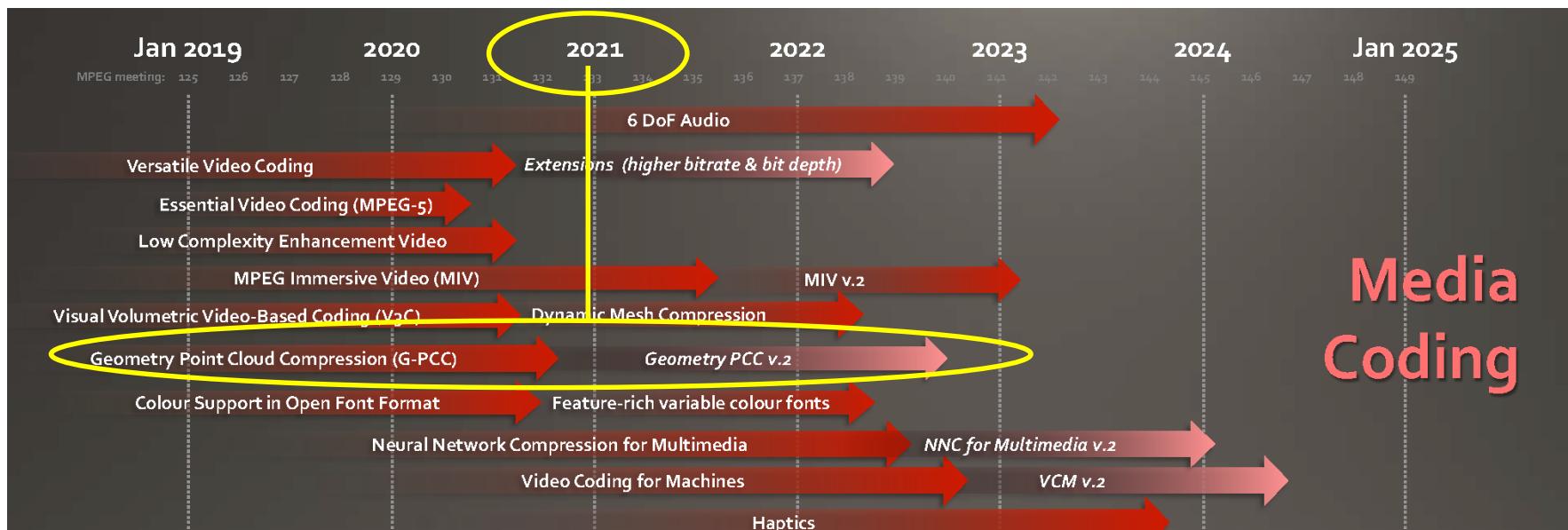
MPEG PCC roadmap

Point Cloud Compression early stages

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

Implementation problems even in local installations

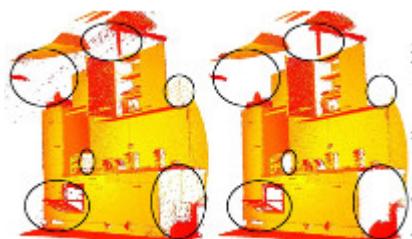
...local network, buses, memory



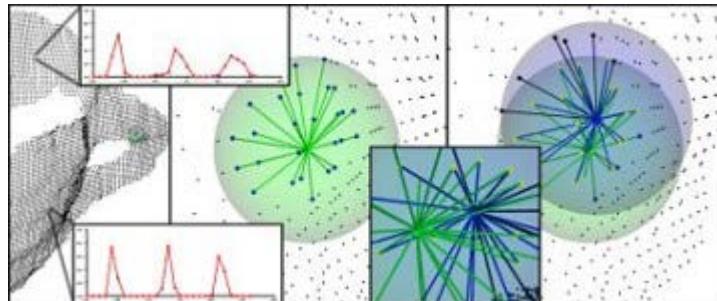
Point Cloud Library (PCL)

- New trend to process raw data produced by scanners

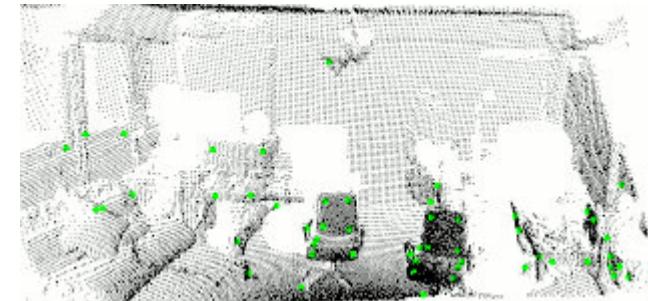
- PCL library B.D Rusu (since 2011)
pointclouds.org v1.11.1 (Jul 2020)
github.com/PointCloudLibrary



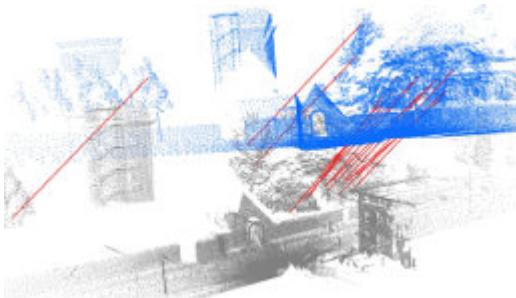
Filters



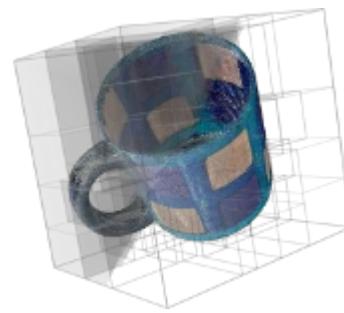
Features



Key points



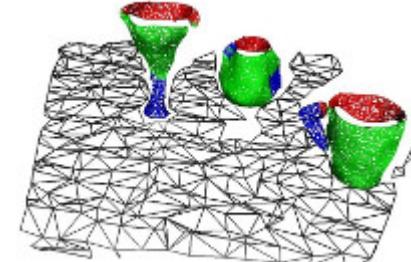
Segmentation



RANSAC



Octree



Visualization

Open3D

- A Modern Library for 3D Data Processing

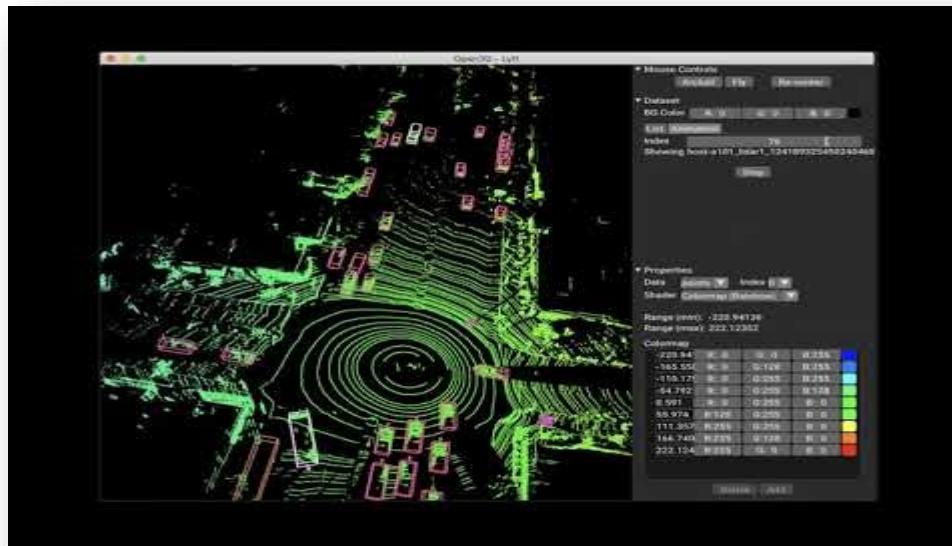
Open3D

Intel ISL (since 2018)

github.com/intel-isl/Open3D

v0.12.0 (Dec 2020)

C++ or Python, Includes PCL



Xavier Suau, Human body analysis using depth data

ORGANIZED POINT CLOUD PROCESSING

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

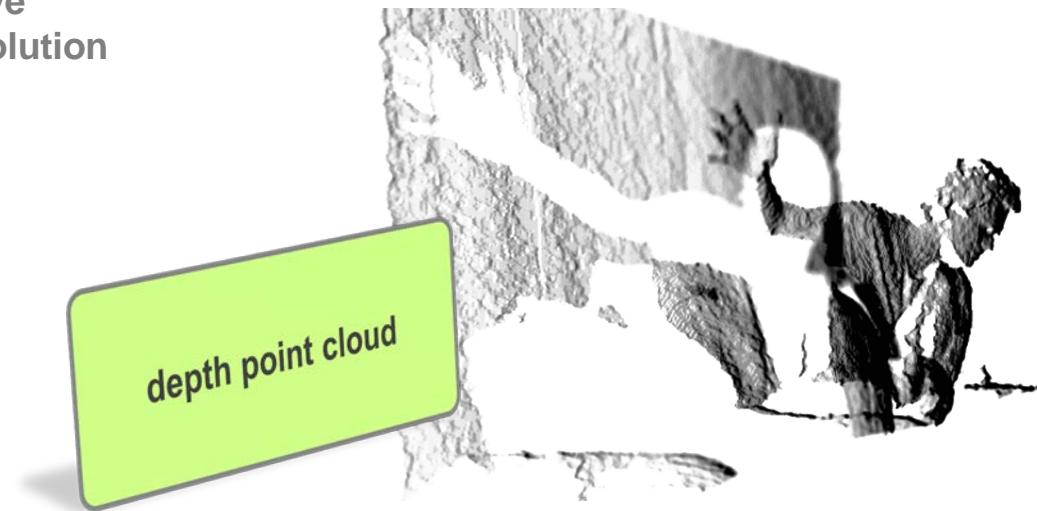


Human Body Analysis using Depth Data



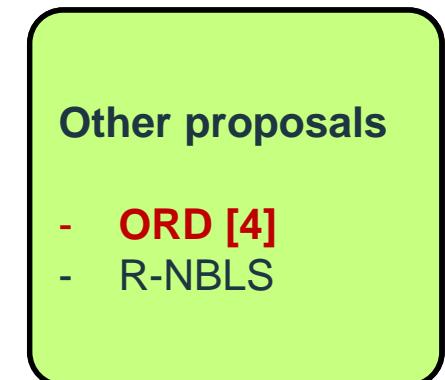
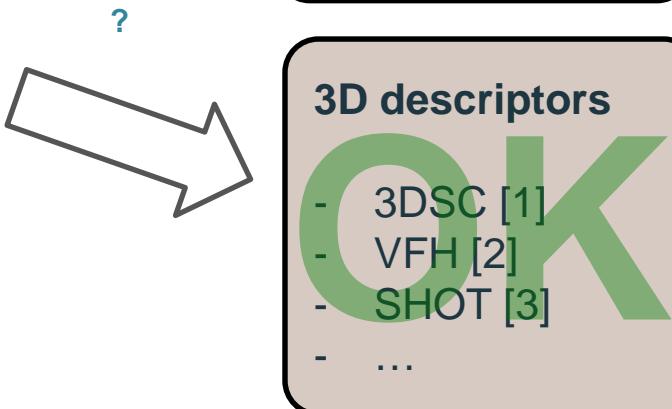
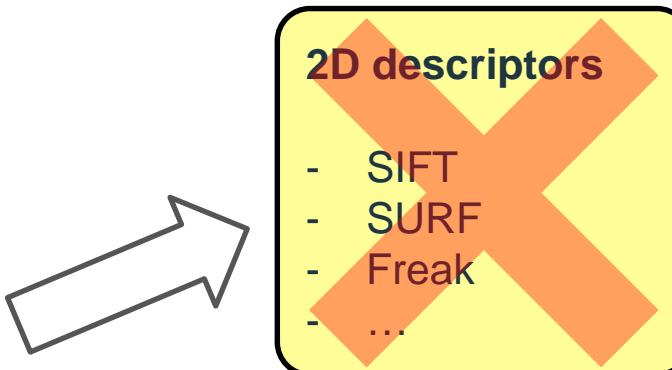
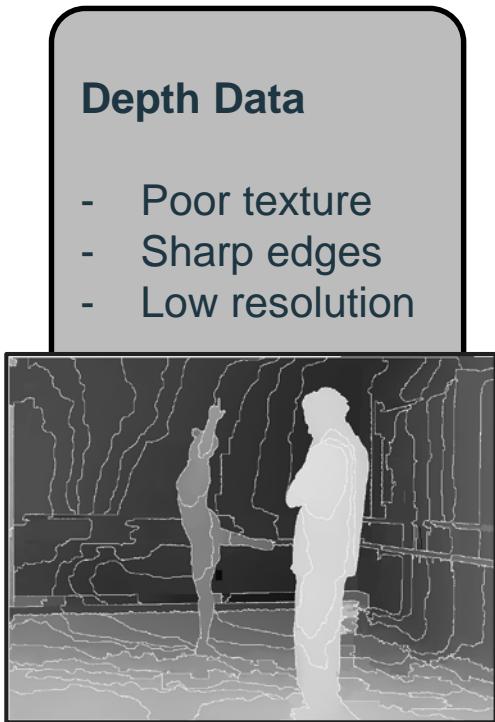
Big
Expensive
Low-resolution

Small
~150€
high-resolution



Depth map / Point Cloud processing

Objective: To obtain information from depth camera frames



[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

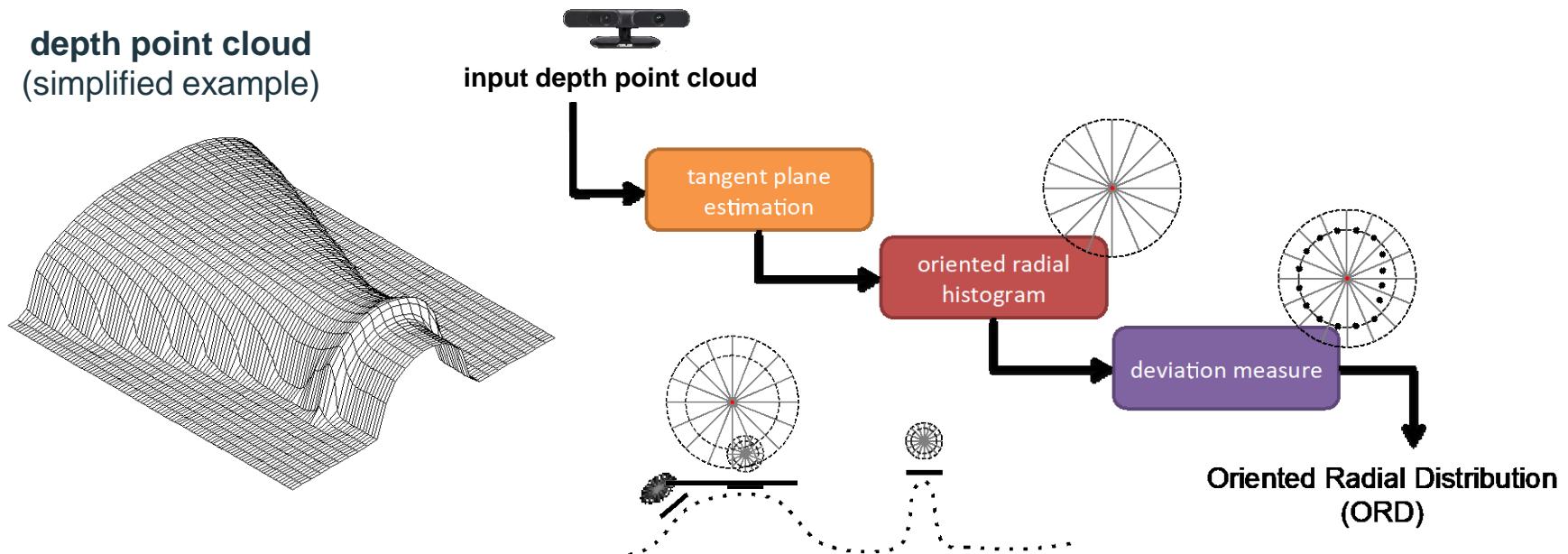
[3] F. Tombari, et al. Unique signatures of histograms for local surface description. ECCV 2010

[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



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



Oriented Radial Distribution

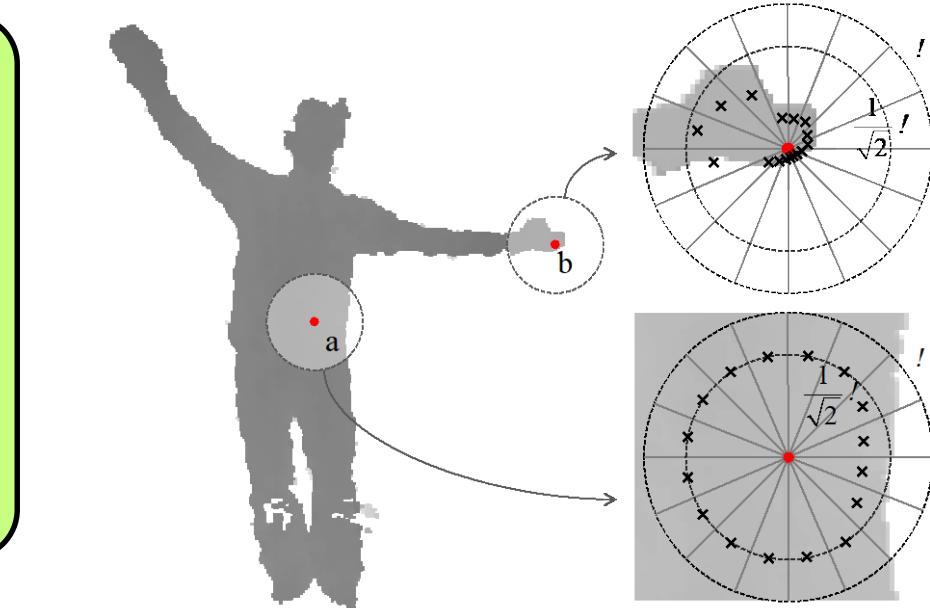
Objective: Detect prominent and flat zones of a depth point cloud

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

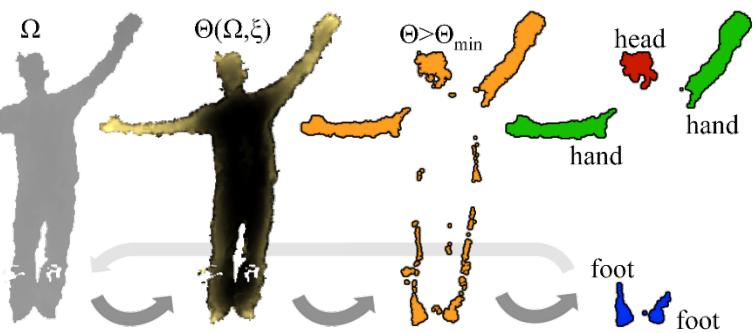


Oriented Radial Distribution

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
 $\gamma_i = \{\text{head, hand, foot, nothing}\}$
depending on its
combined probability

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

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

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 F 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...

Alignment of 3D Point Clouds and RPS Detection

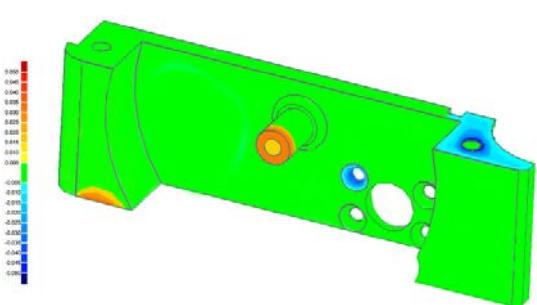
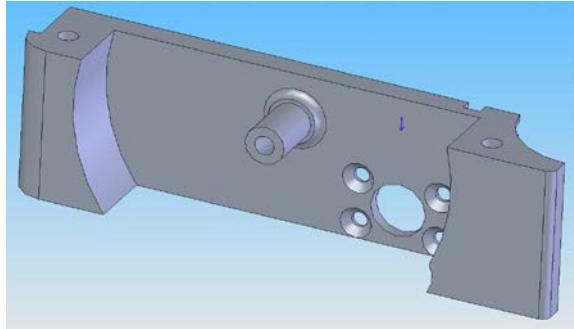
Martin Matilla, ETSETB/UPC, Feb 2015

Industrial (automotive) Production

Computer Aided
Design
(CAD)

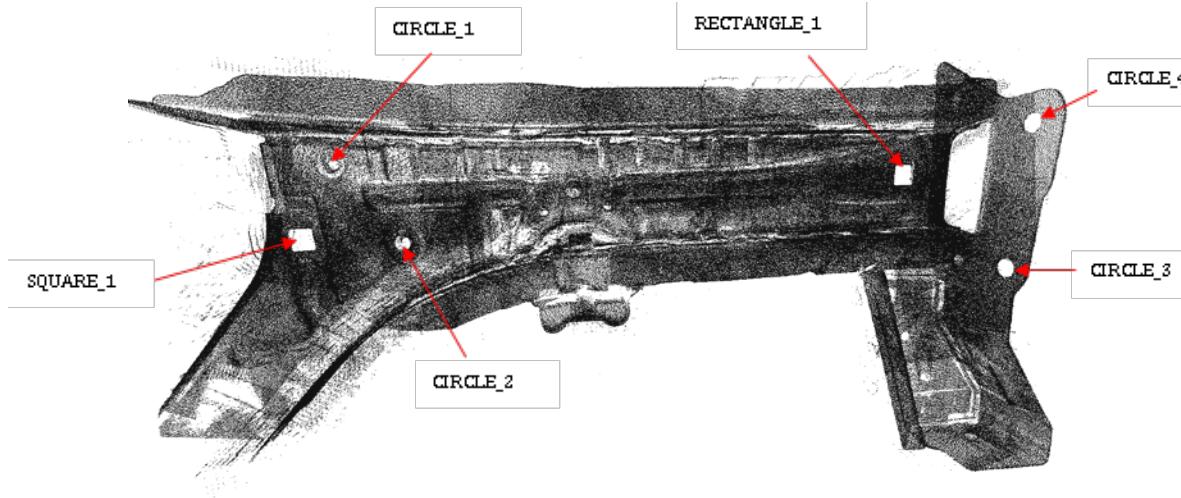
Computer
Numerical
Control (CNC)

Difference
CAD - CNC



Development

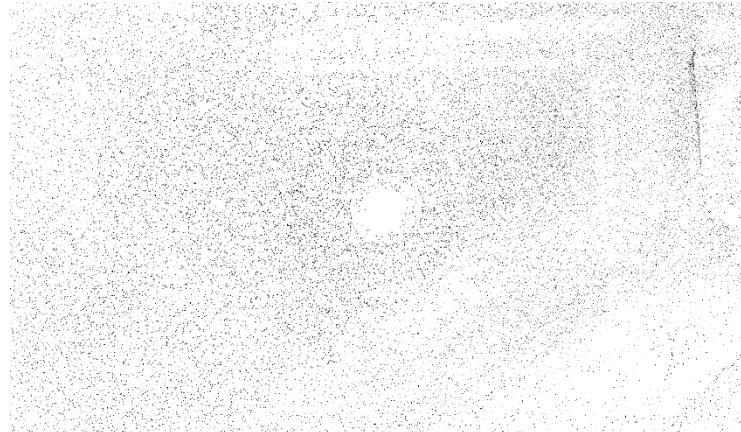
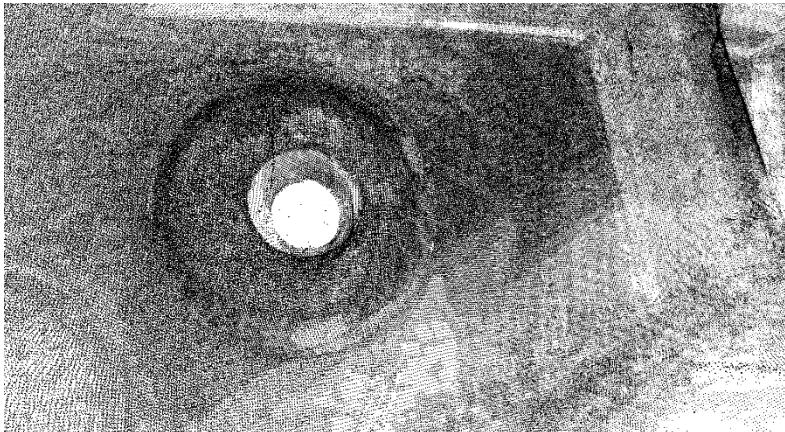
Data



| 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 |

Development

Decimation



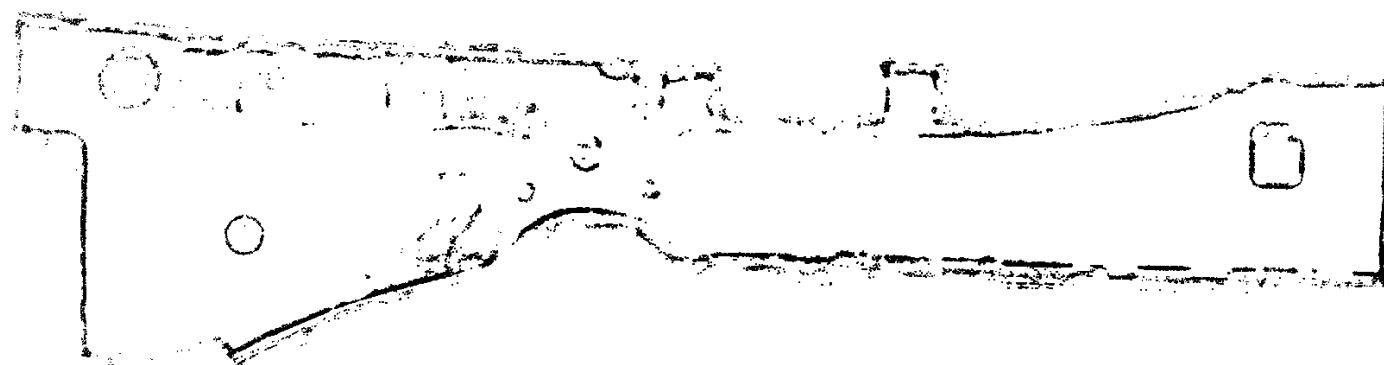
Development

Plane Segmentation



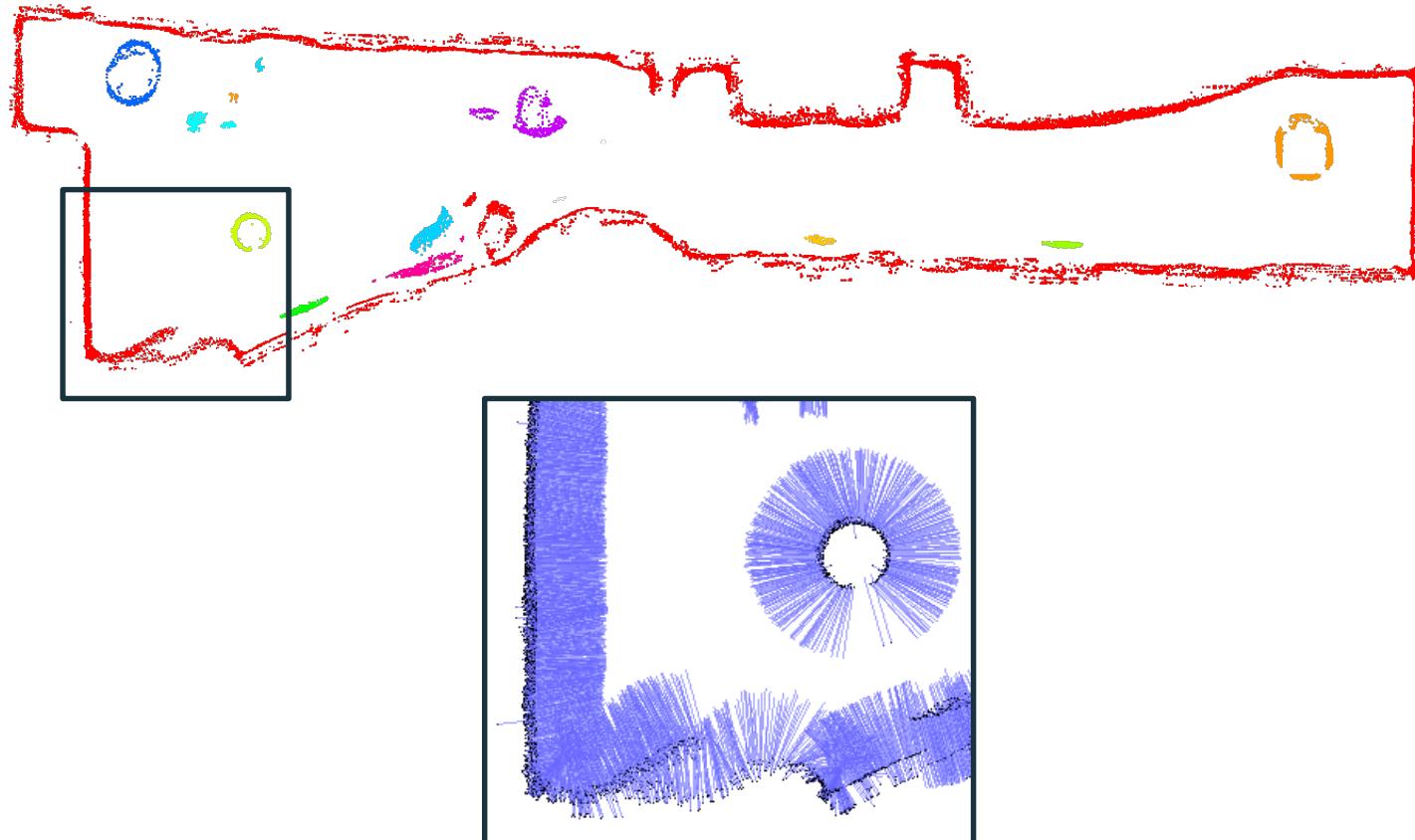
Development

Edge Detection



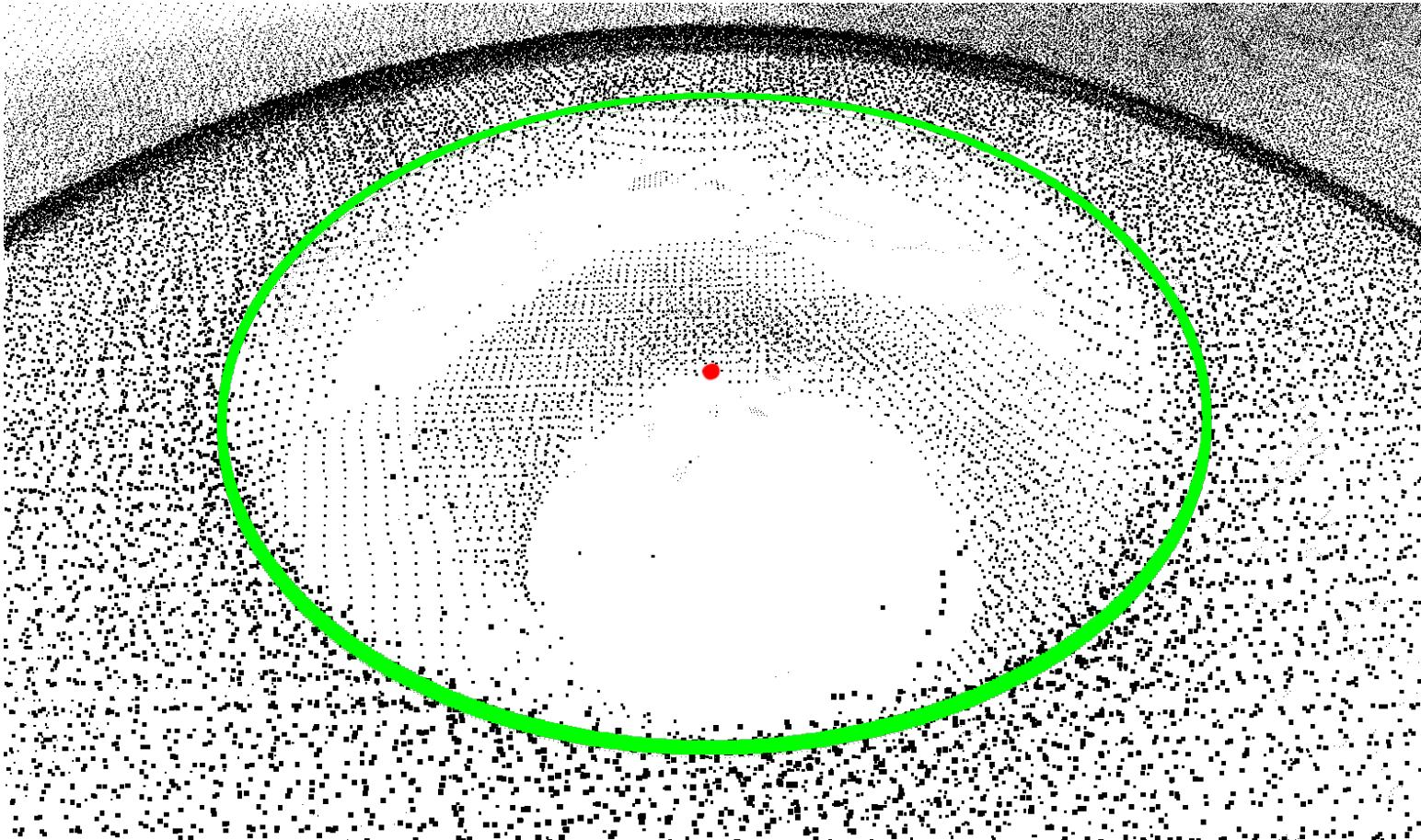
Development

Edge Clustering



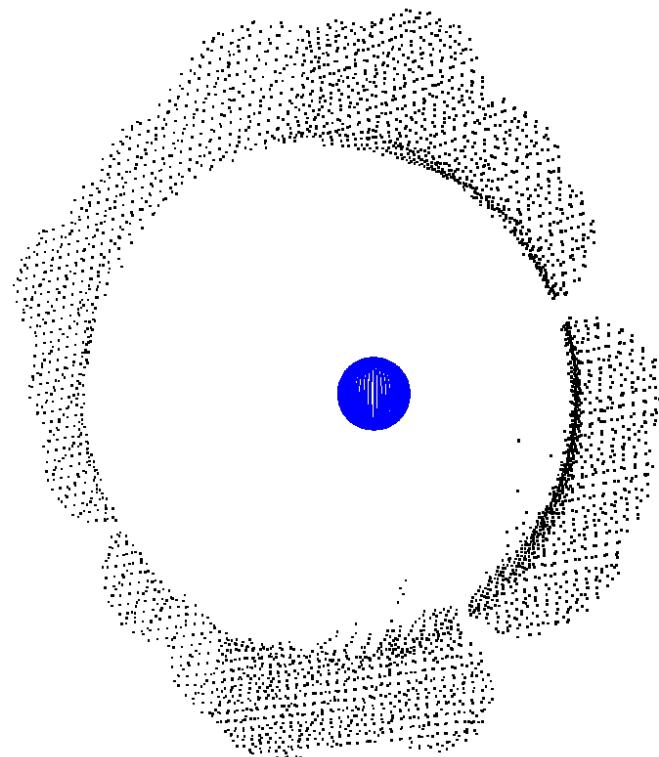
Development

Coarse Detection



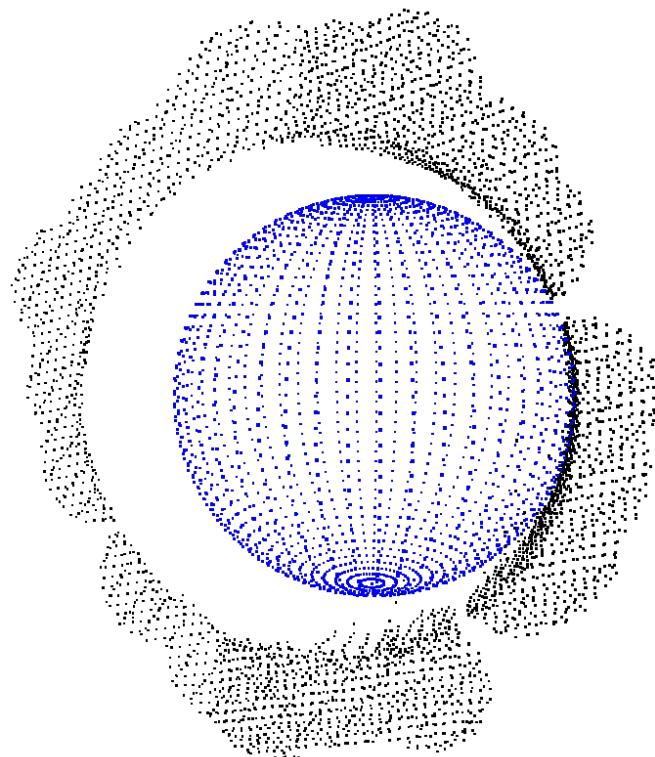
Development

Fine Extraction



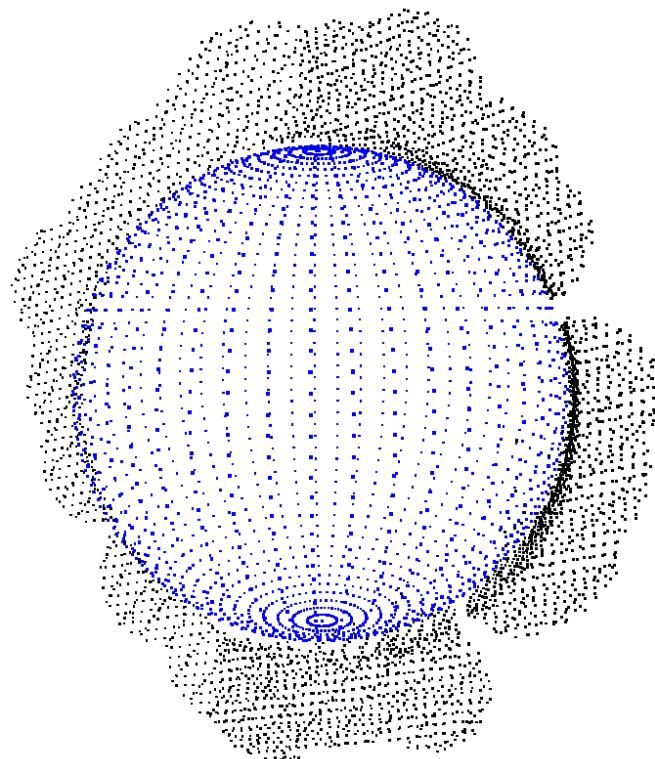
Development

Fine Extraction

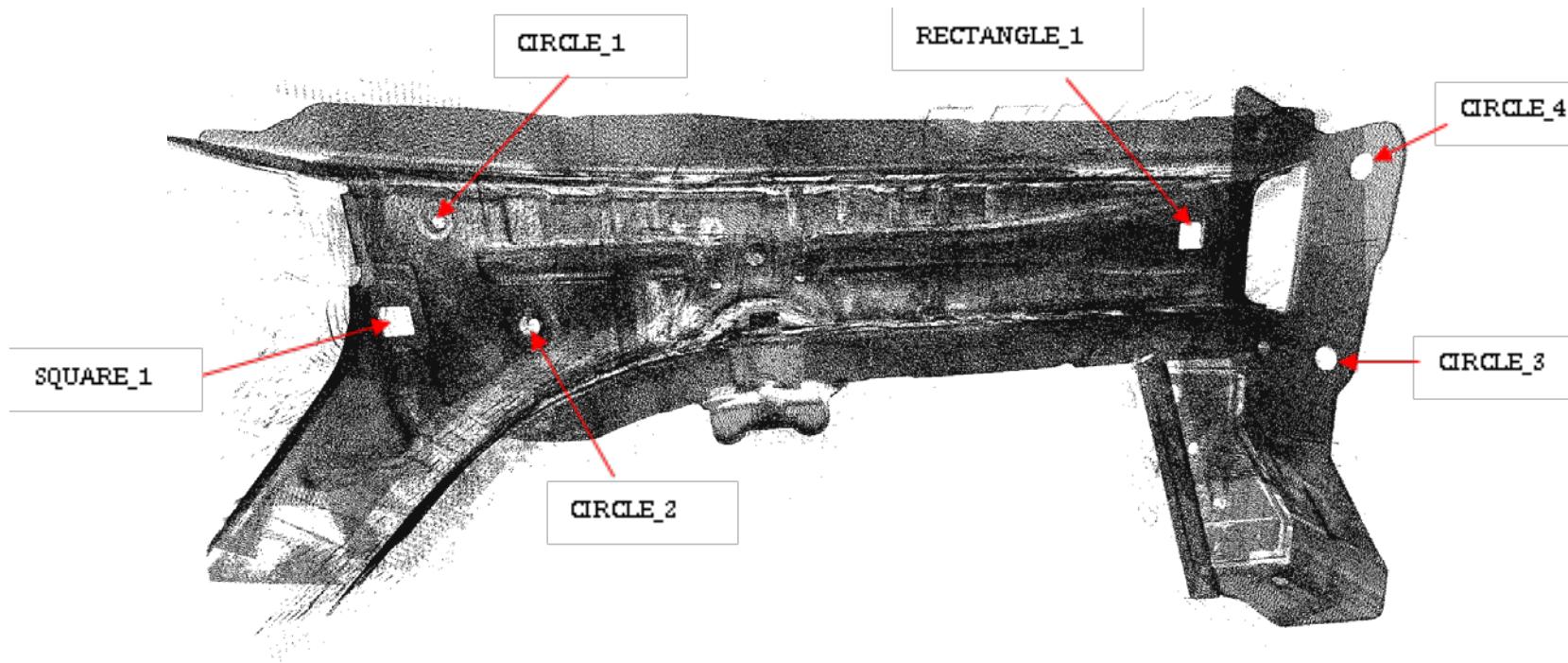


Development

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]



Supervoxels for Stream Data

[Papon 2013]

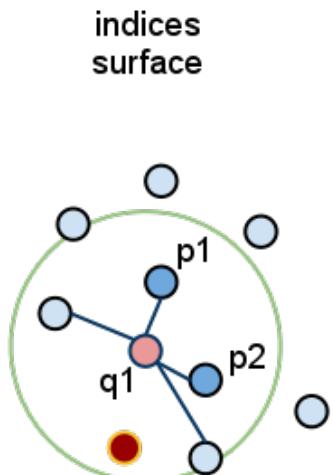
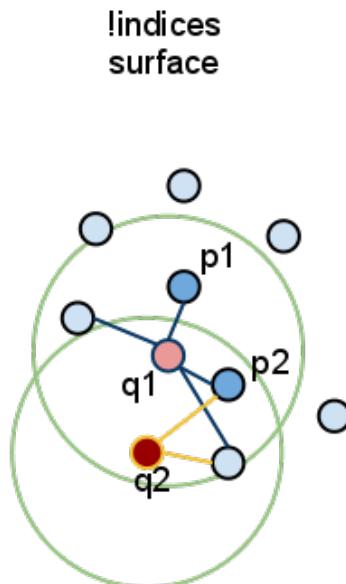
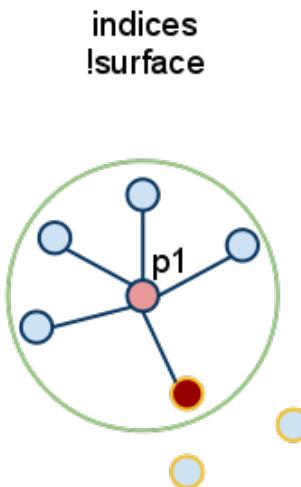
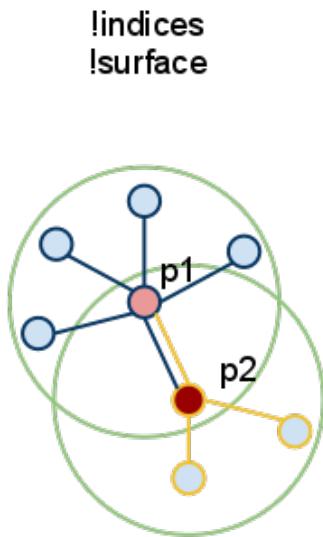


From supervoxels to graphs in HCI applications

- 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:

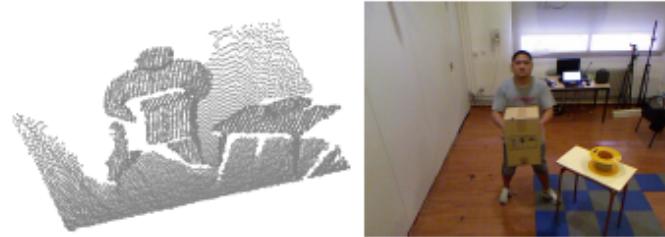
- Often, original data appears unstructured and is based on simple or numerous primitives (pixels, points)
 - Define meaningful primitives, using Graphs or Trees:
 - **Nodes** with meaningful primitives (possibly **heterogeneous**)
 - **Edges** represent relationships... and **hierarchies!**
 - Data 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

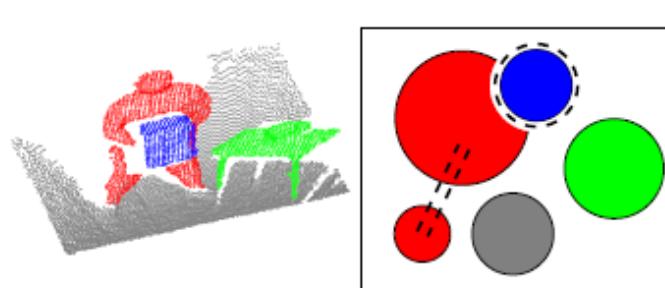
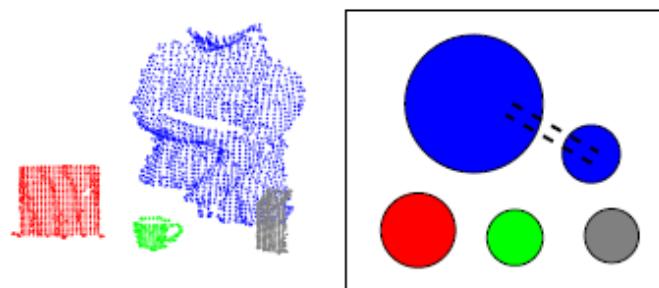
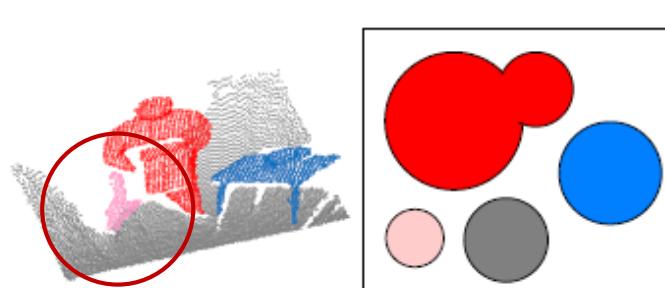
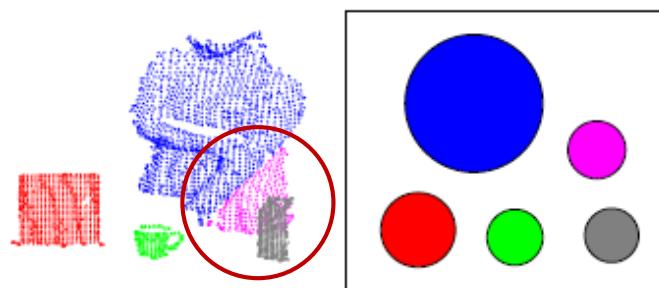
Scene 1



Scene 2



- Point clouds
 - RGB images
-
- Segmentation errors
 - Sketch maps

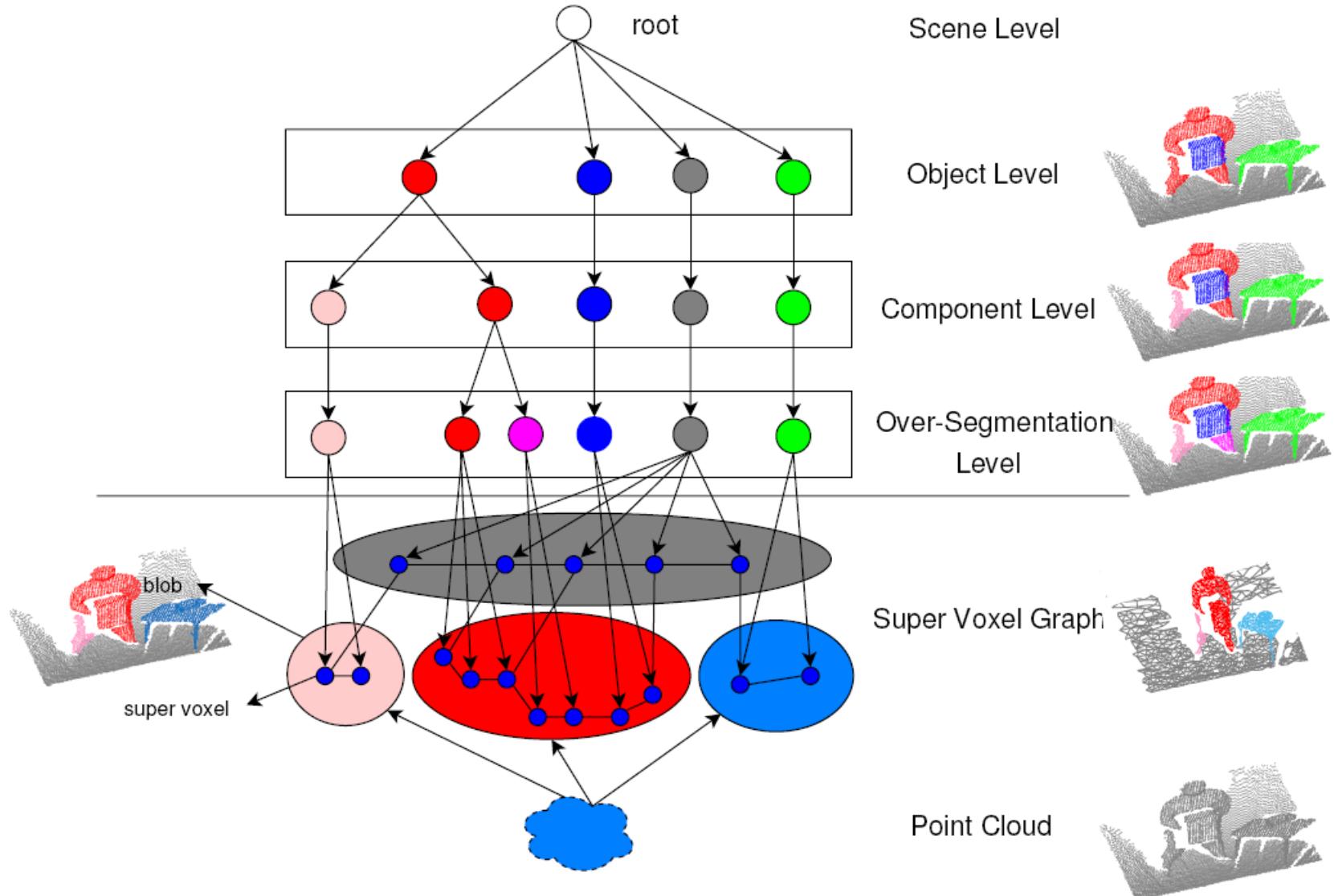


Generic segmentation

Spatio-temporal reasoning from higher level features on trees

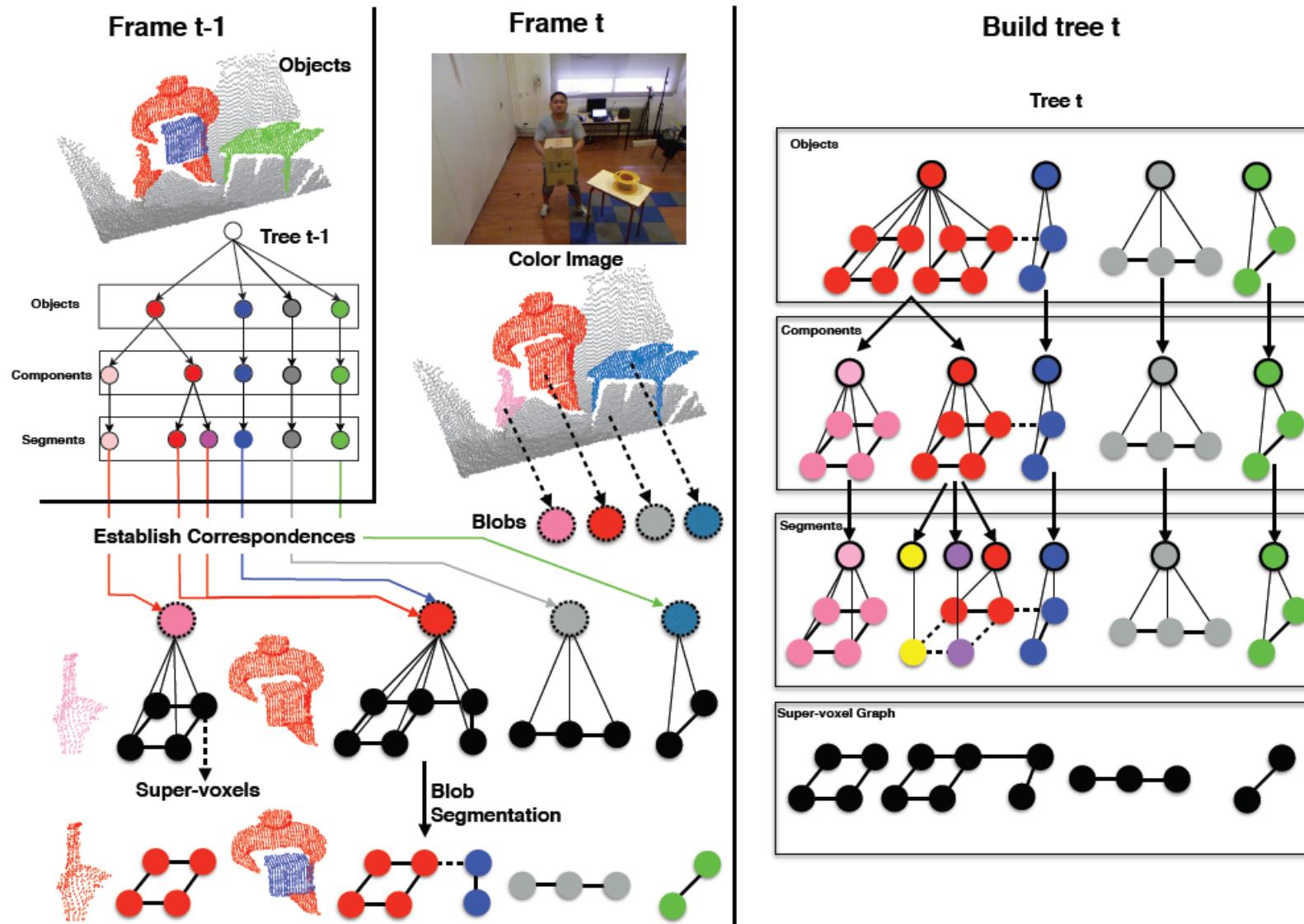
Processing on Trees

[Lin 2018]



Processing on Trees

[Lin 2018]



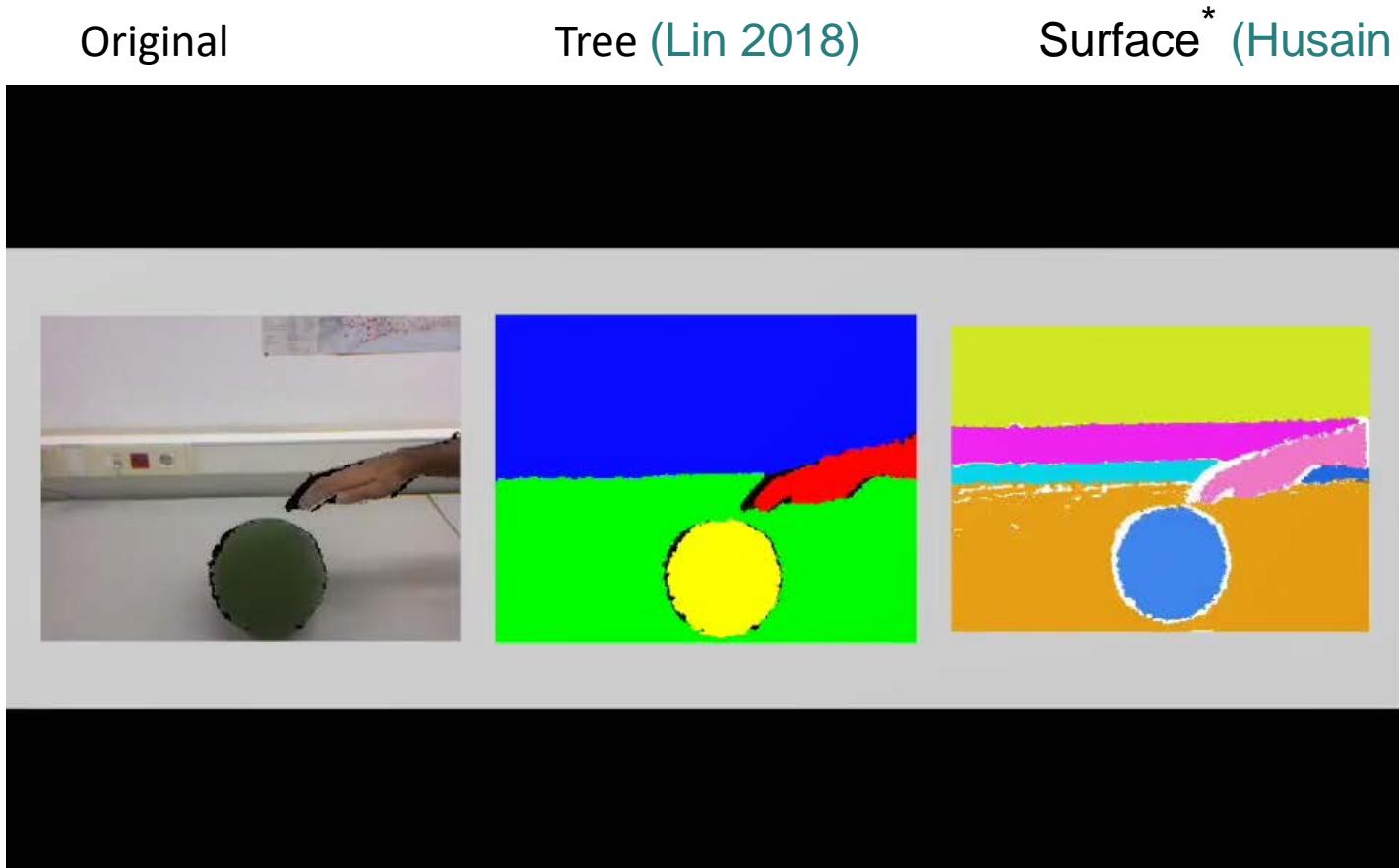
Processing on Trees

[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]



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

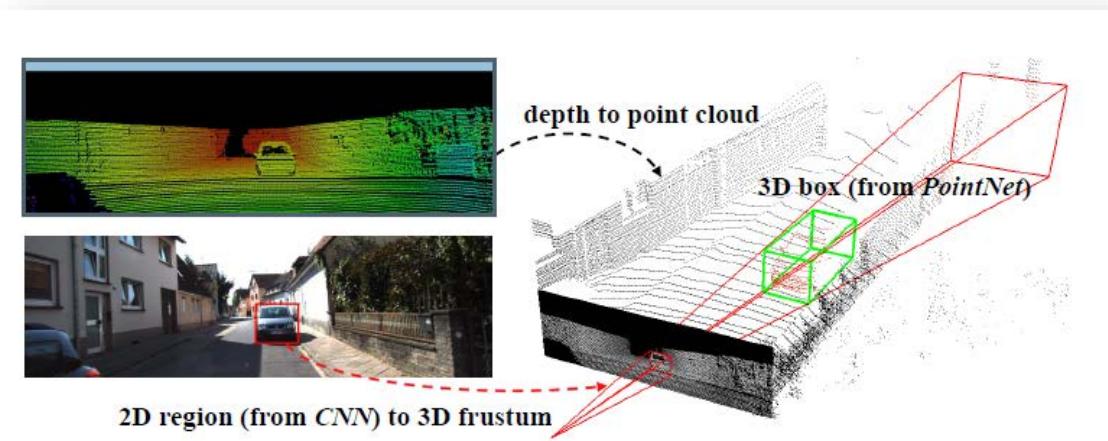
- respects the permutation invariance of points
- each point is processed identically and independently, represented by just its three coordinates (x, y, z)
- additional dimensions added: normals and local/global features
- provides a unified architecture for applications: object classification, part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

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

Frustum PointNet

- Fuses detections in RGB and PointClouds

[Qi 2018]

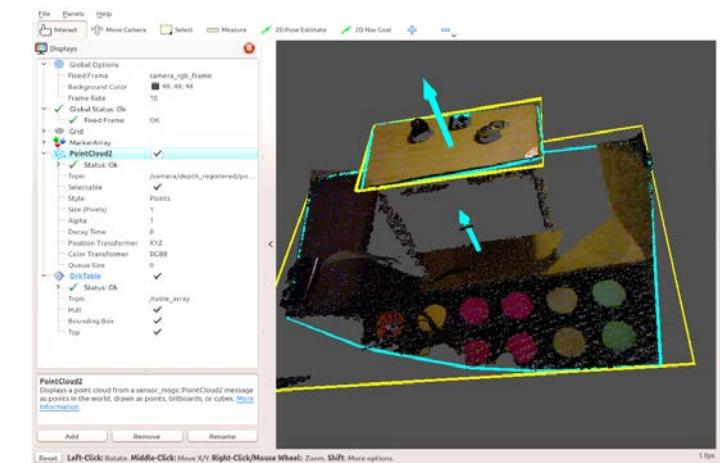
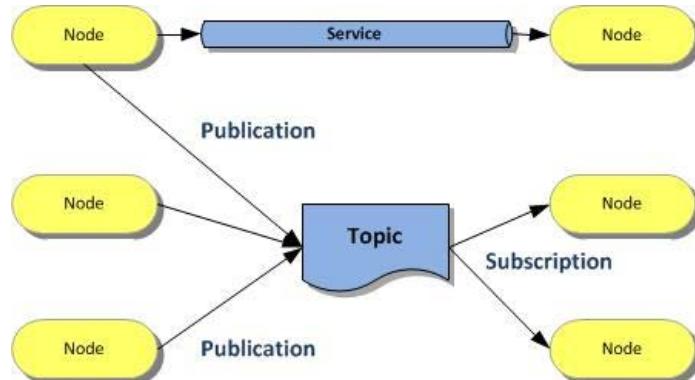


A note on Distributed Processing

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)

ROS



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

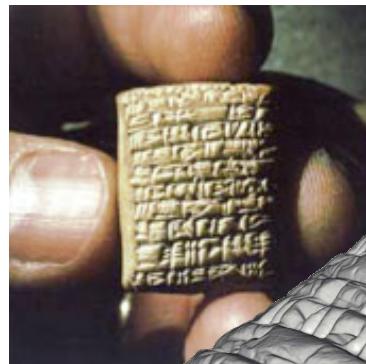
What to do with the results?

MESHING

Motivation for meshing

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

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



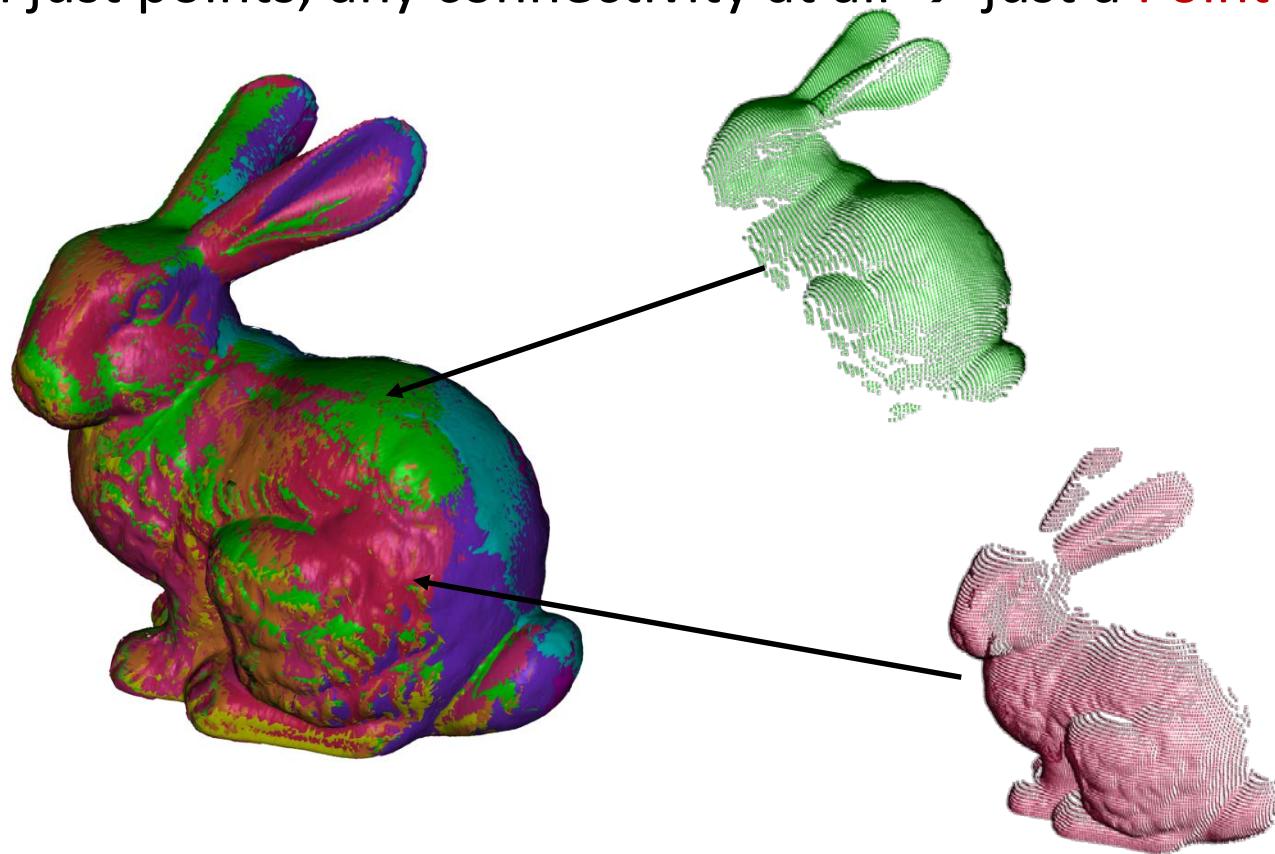
<http://graphics.stanford.edu/projects/mich/>



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

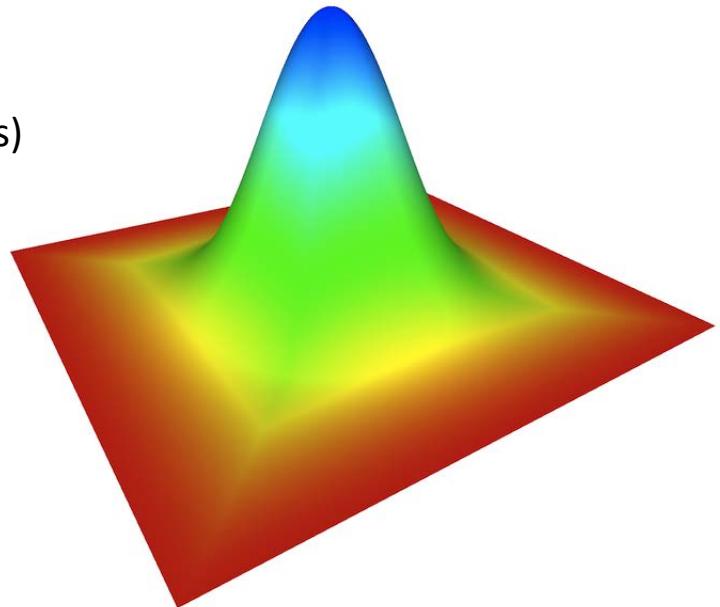
We want a 3D Mesh for:

- 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

Amenta *et al.*, 1998

Edelsbrunner, 1984

Dey *et al.*, 2003

2. Surface Fitting

Terzopoulos *et al.*, 1991

Chen *et al.*, 1995

3. Implicit Function Fitting

Hoppe *et al.*, 1992

Whitaker, 1998

Davis *et al.*, 2002

Turk *et al.*, 2004

Kazhdan, 2005

Curless *et al.*, 1996

Carr *et al.*, 2001

Ohtake *et al.*, 2004

Shen *et al.*, 2004

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:

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

- Computation time
- Memory footprint

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](https://doi.org/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](https://doi.org/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](https://doi.org/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](https://doi.org/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](#)]