Segmentation and Classification of 3D Range Data

Klaas Klasing

Institute of Automatic Control Engineering
Technische Universität München

kk@tum.de

9-Oct-2009, Universidad de Zaragoza
Agenda

1. Introduction
2. Surface-based Segmentation
3. Explaining Occlusions
4. Results
5. Classification
6. Summary & Outlook
Motivation

- Mobile autonomous robotics is gradually moving from 2D navigation towards 3D mapping, interpretation and manipulation
  - Household robotics:
    - Intelligent behavior in domestic setting requires understanding of 3D environments
Active 3D Sensing Hardware

- Actuated 2D laser range finders:
  - SegBot@Stanford
  - ActivMedia
  - UniFreiburg
  - UniHannover

- Time-of-flight cameras:
  - SwissRanger
  - IAS@TUM
  - DualArm@TUM

- Desktop scanning:
  - NextEngine
  - Leica

- Real 3D Lidar:
  - Velodyne

1. Introduction
Range Images vs. Point Clouds

- **Range images** popular in the vision community
  - $z = f(x,y)$
  - powerful algorithms for segmentation
  - problems with projection, depth resolution
  - real 3D scenes must be fused from several 2.5D images

- **3D point cloud**
  - real 3D coordinates (no z-dependency)
  - continuously growing or windowed
  - can contain several registered scans
  - suitable for sweeping 2D range sensors
Ultimate Goal for Domestic Environments

- Quickly and robustly detect objects in 3D environments
  - Point clouds may be sparse, noisy, and may contain occlusions
  - Scanning and interpretation should take only seconds

» Simulator  » Point cloud
Problem Statement

• Consider $n$ points $P = \{p_1, p_2, \ldots, p_n\}$, represented by a data matrix $P = [p_1, \ldots, p_n]^T$, where $p_i = [p_{ix}, p_{iy}, p_{iz}]^T$.

• We’d like to partition the set of points into $n_s$ segments $S_j$, such that

$$S_j = \{p_{j1}, p_{j2}, \ldots, p_{jn_j}\}, \quad n_j > 0 \quad \forall j \in \{1, n_s\}$$

$$S_j \cap S_k = \emptyset \quad \forall j \neq k$$

are disjoint regions of smooth curvature change.
Segmentation of 3D Range Data

- A surprisingly large number of works considers only a few, densely sampled and unoccluded objects in range images:
  - [HJ87], [BJ88], [HJJ96], [B00], [MLM01], [DF02], [BG06]

- Segmentation approaches include
  - Edge-based [B00], [SD01]
  - Curvature-based [BJ88], [HJJ96], [PBJ98]
  - Scanline grouping [JB94], [KMK03]
  - Geometric primitives [MLM01], [BG06], [SWK07]
  - Smooth regions [RHV06], [RMB08]
Normal Vectors

- The two most important features for surface-based segmentation are normal vectors and curvature values.

- For each point $p_i$, a normal vector $n_i = [n_{ix}, n_{iy}, n_{iz}]^T$ is estimated from a set of neighboring points $Q_i = \{q_{i1}, q_{i2}, \ldots, q_{ik}\}$.

- Define a neighborhood matrix $Q_i = [q_{i1}, \ldots, q_{ik}]^T$ and an augmented neighborhood matrix $Q_i^+ = [p_i, q_{i1}, \ldots, q_{ik}]^T$.

- A plethora of different approaches for estimating normal vectors exists [HJ87], [HDD92], [YL99], [GKS00], [HM01], [WTH01], [OF05] ...

For a comparison see
Approaches for Normal Vector Estimation

Optimization-based

$$\min_{n_i} J(p_i, Q_i, n_i)$$

(a) plane fit over neighborhood
(b) angle maximization

(SVD/PCA)

Averaging

$$n_i = \frac{1}{k} \sum_{j=1}^{k} w_j \frac{([q_{i,j} - p_i] \times [q_{i,j+1} - p_i])}{\| ([q_{i,j} - p_i] \times [q_{i,j+1} - p_i]) \|}$$

(c) Average over vectors of neighbor triangles
The neighborhood of each point is defined by a neighborhood graph $G = (V, E)$ with vertices $v_i \in V$ for every point $p_i \in P$ and edges $e_{ij} \in E$.

Two types of graph are used for normal vector estimation:

- **kNN graph**
  - fixed # of neighbors
  - overlapping simplices

- **Delaunay tessellation**
  - variable # of neighbors
  - no overlapping simplices
‘Best‘ Method

- **PlanePCA** \([\text{HDD92}]\)
  
  Minimize the variance of the plane fitting error:

  \[
  \min_{n_i} \left\| \left[ Q_i^+ - \bar{Q}_i^+ \right] n_i \right\|_2
  \]

  where \( \bar{Q}_i^+ \) contains the mean of the points in each row.

- Equivalent to a PCA of \( Q_i^+ \)
  - Principal components = orthonormal basis explaining variance.
  - Variance \( \sim \) eigenvalues of covariance matrix
  - Standard deviation \( \sim \) singular values of data matrix

- Other aspects not addressed here
Curvature Estimation

- Principal curvatures (Euler)
  - Minimal and maximum curvature in orthogonal plane $k_1$ and $k_2$
  - Gaussian curvature: $k_1 \cdot k_2$
  - Mean curvature: $(k_1 \cdot k_2)/2$
- Curvature estimated from paraboloid or cuboid fit [MSR07]
  - estimates too noisy → oversegmentation [HJJ96], [PBJ98]
- Instead, use surface variation: [PGK02], [GHS08]

$$\sigma_{su} = \frac{3\sigma_3}{\sigma_1 + \sigma_2 + \sigma_3}, \quad \sigma_1 > \sigma_2 > \sigma_3$$
Visualization with Curvature

Points grayscale-colored by curvature value $\sigma_{sv} (k=30)$
Smooth Region Segmentation

• Region growing with smoothness constraint proposed in [RHV06]
• Decent results for segmentation of industrial scenes, but
  • segmentation results data dependent
  • sensitive to individual outliers:

• curvature criterion not necessary!
Efficient Segmentation

1. Define two parameters:
   \( \alpha_{max} \): The maximum angle that two neighboring normals may span to belong to the same segment.
   \( \rho_{max} \): The maximum allowed radius of a point neighborhood.

2. For each point calculate the search radius \( \rho_s = \min(\rho_{a,i}, \rho_{max}) \), find the neighbors within this radius and select the ones whose normal vector spans an angle less than \( \alpha_{max} \) with this point’s normal vector.

3. If any of the selected neighbors has been assigned to a segment, assign all selected points to that segment. Otherwise create a new segment and assign all selected points to it. Merge segments if necessary.
Occlusions

- Occlusions cause oversegmentation that can not be remedied by the segmentation algorithm.
- A good segmentation toolchain should be able to explain (basic) occlusions and fix them.
Existing Approaches & Plausibility

• Not many publications on this topic.
• Existing methods can be divided into
  – model-free approaches (hole filling) [DF02], [WO03], [BPB06]
  – model-based approaches (object recognition) [B00], [BI00], [TB07]

• Important: Reason about plausibility of occlusion [DF02]
Merging Oversegmented Surfaces

- Identify candidate surfaces by 4 distance measures

- Identify pairs of boundary points on each segment:

3. Explaining Occlusions
Checking for Occlusion

• Check connecting lines for occlusion
  – Create an omnidirectional range image (ORI) for all points observed from one view point:

\[
a_i = \left\lfloor \left( \frac{1}{2\pi} \text{atan2}(p_{iy} - v_{iy}, p_{ix} - v_{ix}) + 0.5 \right) \right\rfloor w
\]

\[
e_i = \left\lfloor \left( \frac{1}{\pi} \text{atan2}(p_{zi} - v_{zi}, \sqrt{(p_{ix} - v_{ix})^2 + p_{iy} - v_{iy}^2}) + 0.5 \right) \right\rfloor h
\]

\[
d_i = ||p_i - v_i||_2
\]

– For each boundary line check the corresponding line in the image for occlusion
4. Results
Experiment

4. Results
Results - Experiment

4. Results
Discussion

• The bulk of the processing time is spent on normal vector estimation
  – Good normal estimates crucial for high segmentation quality
  – Runtime complexity linear in $k$
  – Possible solution: parallelize estimation

• Current limitation:
  – View points required to explain occlusions
    → Extend this to view point trajectories
Classification

• Goal: use the segmented surfaces as an abstract representation for detecting objects

• Segment features:
  – Centroid, normal vector, surface variation, axis-aligned bounding box, convex hull etc.
Possible Methods/Frameworks

• Train primitive classifiers for parts SVMs, Decision Trees, NeuralNets etc. trained from segment features
• Find structure in neighboring classified segments
  – convert neighborhood relationship to graph
  – complete/sub-graph matching
  – part-based approaches [RDR95], [GHS08]
• Train with IKEA CAD models → Recognition in real data sets?
Conclusion

• Efficient segmentation toolchain
• Applicable to
  • sparse and noisy point clouds
  • variable density data, registered views, partial occlusions
• Segmentation time on the order of the scan time
• Segments provide informative features for classification

• **Central question:** most efficient/robust framework for classification?

---

This work is supported in part within the DFG excellence initiative research cluster Cognition for Technical Systems – CoTeSys, see also www.cotesys.org.
References


References


