Primitive-based surface reconstruction

Florent Lafarge
Inria Sophia Antipolis - Mediterranee
- **Geometric primitive extraction**
  - Region growing
  - Ransac
  - Accumulation methods
  - Global regularities

- **Surface reconstruction using geometric primitives**
- **Two words on template matching**
Why Geometric primitives can be interesting for surface reconstruction?

- High complexity
- No structure

Smooth reconstruction
Why Geometric primitives can be interesting for surface reconstruction?
Why Geometric primitives can be interesting for surface reconstruction?
Why Geometric primitives can be interesting for surface reconstruction?
How to extract Geometric primitives from point sets?
Region growing

- Iterative method
- Spatial propagation of a primitive Hypothesis
- deterministic
- Efficient for relatively “clean” Data
Region growing

- select a point and a primitive hypothesis
Region growing

- select a point and a primitive hypothesis
- propagate to the neighbors if they verify the hypothesis
Region growing

- select a point and a primitive hypothesis
- propagate to the neighbors if they verify the hypothesis, and iterate until no point verifies the hypothesis anymore.
Region growing

- select a point and a primitive hypothesis
- propagate to the neighbors if they verify the hypothesis, and iterate until no point verifies the hypothesis anymore.
Region growing

- select a point and a primitive hypothesis
- propagate to the neighbors if they verify the hypothesis, and iterate until no point verifies the hypothesis anymore.
Region growing

- select a point and a primitive hypothesis
- propagate to the neighbors if they verify the hypothesis, and iterate until no point verifies the hypothesis anymore.
Region growing

- select a point and a primitive hypothesis
- propagate to the neighbors if they verify the hypothesis, and iterate until no point verifies the hypothesis anymore.
- select a remaining point and a primitive Hypothesis, and iterate
Region growing

- select a point and a primitive hypothesis
- propagate to the neighbors if they verify the hypothesis, and iterate until no point verifies the hypothesis anymore.
- select a remaining point and a primitive Hypothesis, and iterate
the parameters to specify

- minimum number of points needed to fit the primitive
- Distance threshold
Region growing

- need to know the nearest neighbors
- the primitive hypothesis has to be relevant when starting the growing
- .. but the primitive hypothesis can also be updated during the growing
- not optimal when noisy data
Region growing

using normals

using Euclidian distance

using normals and Euclidian distance
Ransac (RANdom SAmple Consensus)

- Iterative method
- Estimation of the primitive parameters by a random sampling of data
- Designed to be efficient with outlier-laden Data
- Non-deterministic
Ransac Algorithm

- Sample (randomly) the number of points required to fit the primitive
- Solve for primitive parameters using samples
- Score by the fraction of inliers within a preset threshold of the primitive

Repeat these 3 steps until the best primitive is found with high confidence
Ransac Algorithm

- Sample (randomly) the number of points required to fit the primitive

- Solve for primitive parameters using samples

- Score by the fraction of inliers within a preset threshold of the primitive

Repeat these 3 steps until the best primitive is found with high confidence

[Fischler & Bolles]
Ransac Algorithm

- Sample (randomly) the number of points required to fit the primitive

- Solve for primitive parameters using samples

- Score by the fraction of inliers within a preset threshold of the primitive

Repeat these 3 steps until the best primitive is found with high confidence

[Fischler & Bolles]
Ransac Algorithm

- Sample (randomly) the number of points required to fit the primitive

- Solve for primitive parameters using samples

- Score by the fraction of inliers within a preset threshold of the primitive

Repeat these 3 steps until the best primitive is found with high confidence

\[ N_I = 6 \]
Ransac Algorithm

- Sample (randomly) the number of points required to fit the primitive
- Solve for primitive parameters using samples
- Score by the fraction of inliers within a preset threshold of the primitive

Repeat these 3 steps until the best primitive is found with high confidence

\[ N_I = 14 \]
the parameters to specify

- minimum number of points needed to fit the primitive
- Distance threshold \( \delta \)
- Number of samples
  To be chosen so that at least one random sample is free from outliers with a certain probability

[Fischler & Bolles]
Accumulation methods

- Accumulate local primitive hypotheses in a space of primitive parameters
- extract the local maxima from the parameter space
- the parameter space must be discretized
Accumulation methods: Hough transform

Case of lines in 2D

(Hough, 1959)
Accumulation methods: Hough transform
Accumulation methods: Gaussian sphere

For each point of the data, we increment the sphere cell targeted by the point normal from the sphere center.
Accumulation methods: Gaussian sphere

For each point of the data, we increment the sphere cell targeted by the point normal from the sphere center.
Accumulation methods: Gaussian sphere

An accumulation of points in the Gaussian sphere
Allows the detection of one or several planes with a similar orientation
Accumulation methods: Gaussian sphere

An accumulation of points along a circle in the Gaussian sphere allows the detection of one or several cylinders with a similar orientation.
Accumulation methods

- can be computationally expensive
- restricted to certain types of primitives
- can be interesting for “structuring” the primitive configuration with global regularities
Global regularity discovering

- Equal length
- Parallel faces
- Coplanar
- Orthogonal faces
- Equal angle

[Li et al., 2011]
Global regularity discovering

- usually primitives are detected locally, without interaction between each others

- It can be useful to introduce interactions between primitives at a global scale

[Li et al., 2011]
Global regularity discovering [Globfit]

[Li et al., 2011]
Global regularity discovering [Globfit]

[Li et al., 2011]
- Geometric primitive extraction
- Surface reconstruction using geometric primitives
  - Graph-based
  - Space partitioning
  - Hybrid reconstruction
- Two words on template matching
Surface reconstruction from geometric primitives

Q: What can we do once we have extracted the primitives?
A1: compute the primitive adjacency graph, and reconstruct the surface as the dual of this graph.
A1: compute the primitive adjacency graph, and reconstruct the surface as the dual of this graph.

If you are lucky..
A1: compute the primitive adjacency graph, and reconstruct the surface as the dual of this graph.

Ideal case: this never happens in practice

- No guarantee of finding the right primitive configuration and right adjacency graph
- No guarantee that the observed scene can be entirely explained by geometric primitives
A2: Use primitives to partition the space into cells to be labeled as inside or outside

[Labatut et al., 2009]
- works well when no missing primitive

[Labatut et al., 2009]
- when primitives are missed or cannot be detected, use of ghost primitives

[Chauve et al., 2010]
when primitives are missed or cannot be detected, use of ghost primitives

[Chauve et al., 2010]
when primitives are missed or cannot be detected, use of ghost primitives
- A3: reconstruct an hybrid surface as a combination of canonical parts idealizing the primitives and free-form parts representing the smooth or undetected canonical elements.
Hybrid reconstruction by structuring

Starting from a point set and a configuration of planar primitives extracted under a tolerance $\varepsilon$

- 2-manifold
- watertight
- intersection-free

[Lafarge et Alliez]
- 3 ideas
- 3 ideas
  - Meaning insertion
- **3 ideas**
  - Meaning insertion
  - Structure idealization under Delaunay triangulation
- 3 ideas
  - Meaning insertion
  - Structure idealization under Delaunay triangulation
  - Complexity reduction

[Image: Diagram showing 233K pts and 44K pts]
Replacement of the inliers by an *ideal* layout of planar points

- Occupancy 2D-grid projected on the planar primitive
Replacement of the inliers by an *ideal* layout of planar points

- Occupancy 2D-grid projected on the planar primitive
- Facet existence condition in Delaunay: \( L_p < \sqrt{2} \varepsilon \)

[Lafarge et Alliez]
Preservation of edges between adjacent primitives

- Occupancy 1D-grid projected on the intersection line
- Facet existence condition in Delaunay:

\[
\begin{align*}
L_c &= 2\varepsilon \\
h_c &= \varepsilon \times \cos \frac{\theta}{2}
\end{align*}
\]
- **Corner points**
  added by detecting the potential n-cycles extracted from the detected 3-cycles from the primitive

- **Clutter points**
  correspond to the input points which have not been detected as belonging to planar primitives
Space partitioning: 3D delaunay triangulation from the structured point set

- tetrahedra do not intersect the primitive-induced surfaces
- each vertex of the triangulation inherits from a structural type

Structured point set
0.08M points

0.51M cells
1.02M facets
Labeling the Delaunay cells

- a graph \((C,F)\)
  
  \[ C = \{c_1, \ldots, c_n\} \] the set of Delaunay cells
  
  \[ F = \{f_1, \ldots, f_m\} \] the set of triangular facets separating two cells

- a cut \((C_{in}, C_{out})\) in the graph
  
  The set of facets separating \(C_{in}\) from \(C_{out}\) forms a surface \(S\)

- a cost function \(C\) measuring the quality of a cut
  
  \[
  C(S) = \sum_{f_i \in S} a(f_i) Q(f_i) + \sum_{c_k \in C_{in}} P_{out}(c_k) + \sum_{c_k \in C_{out}} P_{in}(c_k)
  \]

  Geometric quality

  Visibility prediction

- an optimization algorithm for finding the optimal cut [Boykov2004]
Visibility prediction

- detection of visibility patches by ray shooting
- *inside/outside* prediction of Delaunay cells crossed by a ray

\[
\begin{align*}
P_{\text{out}}(c_k) &= \beta \cdot 1_{\{c_k \in \mathcal{P}_{\text{out}}\}} \\
P_{\text{in}}(c_k) &= \beta \cdot 1_{\{c_k \in \mathcal{P}_{\text{in}}\}}
\end{align*}
\]
Visibility prediction
- detection of visibility patches by ray shooting
- *inside/outside* prediction of Delaunay cells crossed by a ray

[Image: Diagram showing input point set, spatial density of points, and Delaunay triangulation with predicted OUT and no prediction regions.]

[Statistical data: original point set (2M pts), 56.5K patches, 134.2K cells predicted IN; 23.7K patches, 83.9K cells predicted IN; +20% outliers, +0.2% noise.]

[Reference: Lafarge et Alliez]
Geometric quality

- **S-coherent facets**
  Plausible facets as a portion of a canonical part

- **FF-coherent facets**
  Plausible facets as a portion of a freeform shape.

- **Incoherent facets**
  all the remaining cases

\[
Q(f_i) = \begin{cases} 
0 & \text{if } f_i \text{ S-coherent} \\
g(f_i) & \text{if } f_i \text{ FF-coherent} \\
\gamma & \text{if } f_i \text{ incoherent}
\end{cases}
\]
Surface simplification: edge-collapse exploiting the structural meaning of vertices

- canonical parts edge length cost to edges linking identical planar or crease vertices
- free-form parts Keep unchanged
Hybrid vs smooth

- Extracted primitives
- Our hybrid reconstruction
- Poisson reconstruction

Hausdorff distance to input points

≥1% Bbox diagonal
Hausdorff distance to input point set ( % bbox diagonal)

Time (s)

ε (% bbox diagonal)

Blade
Hausdorff distance to input point set (\% bbox diagonal)

Church

smooth [Kazhdan et al. 2006]
Shape approximation [Cohen-Steiner et al 2004]
Piecewise-smooth [Salman et al. 2010]
Interactive primitive [Arikan et al. 2012]
 Geometric primitive extraction
 Surface reconstruction using geometric primitives
 Two words on template matching
Template matching

- Geometric primitives are usually simple, eg planes or cylinders
- But sometimes, we need to fit complex primitives to the data..
Problems

- Do we search for one or several objects in the data?
- If several, do we know the number of objects?
- Can objects interact between each others?
Here, we don’t know the number of objects and interactions must be inserted (spatial overlapping, tree competition..)

.. this is not surface reconstruction anymore