Point Cloud Structural Parts Extraction based on Segmentation Energy Minimization

B. Cafaro, I. Azim, V. Ntouskos, F. Pirri, M. Ruiz

GRAPP 2015 - 14/03/2015 - Berlin, Germany
Overview

• Problem description
• Related work
• Preliminaries
  • voxel space
  • level-sets
• Feature extended distance functions
• Motion models
• Implementation
• Results
• Conclusions
Problem description

Segment and reconstruct structural parts of a scene based on geometric features, estimated from an unorganized point cloud (PCL)

Our approach:
Level-set method guided by a distance function incorporating geometric features
## Related work

### Surface segmentation methods

<table>
<thead>
<tr>
<th>Unorganized PCL based (Nguyen and Le, 13)</th>
<th>Mesh based (Shamir, 08)</th>
</tr>
</thead>
<tbody>
<tr>
<td>• edge based</td>
<td>• region growing</td>
</tr>
<tr>
<td>• attribute based</td>
<td>• hierarchical clustering</td>
</tr>
<tr>
<td>• graph based</td>
<td>• iterative clustering</td>
</tr>
<tr>
<td>• region based</td>
<td>• spectral analysis</td>
</tr>
<tr>
<td>• model based</td>
<td>• implicit methods</td>
</tr>
</tbody>
</table>
Voxel space

- Voxel space defined by PCL bounding box
- Padding to allow for level-set evolution
- Voxel size defined by PCL mean density
- PCL filtered according to point density variance
- Each voxel contains up to one point
Level-set

Level-set equation:

\[ \phi_t(t) + v \cdot \nabla \phi(t) = 0 \]

for internally generated velocity field, evolution follows the normal:

\[ \phi_t(t) + u || \nabla \phi(t) || = 0 \]

Search the surface with minimum energy:

\[ E(\Gamma) = \int_{\Gamma} (\rho(S, w))^2 \ ds \]

with \( \rho(S, w) = \min_{x_i \in S} (||w - x_i||) \) and \( \Gamma \) the 0 level-set of \( \phi \)

Initialization \( \phi(w, 0) \) is also required
Feature extended distance function

At each point, assign a descriptor:

\[ F(x_i) = \{K, h, n\}, \]

with \( K \) mean curvature, \( h \) height, \( n \) normal

At each voxel center \( m_{i,j,k} \), assign the descriptor of its nearest neighbor:

\[ F(m_{i,j,k}) = F(\mathcal{N}(m_{i,j,k})) \]

Transformed distance function:

\[
\rho_F(S, w) = \begin{cases} 
\rho(S, w) + H & \text{if } F \neq F^* \\
H & \text{otherwise} 
\end{cases}
\]

with \( H = c_1|K - K^*| + c_2|h - h^*| + c_3(\arccos(n^Tn^*)) \)
Motion model $\mathcal{M}_1$

Model defined by:

- $\rho_F(S, w) = \rho(S, w) + H$,
- $u = \gamma_\Omega(\mathcal{B})$,
- $\phi(w, 0) = || w - c || - r$

with $\mathcal{B} = 1_{\rho_F(S, w) > t}$ and $\gamma_\Omega(\mathcal{B}) = DT(\mathcal{B}) + DT(1 - \mathcal{B}) + 0.5\mathcal{B}$

- $\rho_F$ always positive, but not smooth (only near points)
- global minimum of $\rho_F$ at the voxels which contain the structural part to be segmented
- initialize with general shape (sphere)
Motion model $\mathcal{M}_2$

Model defined by:

- $\rho_F(S, w) = \rho(S, w) + H$,
- $u = \rho_F(S, w)$,
- $\phi(w, 0) = \gamma_\Omega(B) - b$

with $B = 1_{\rho_F(S, w) > t}$ and $\gamma_\Omega(B) = DT(B) + DT(1 - B) + 0.5B$

- $\rho_F$ used also for the evolution velocity
- initialization based on $\gamma_\Omega(B)$
- convergence only if $b$ is suitably selected
Motion model $\mathcal{M}_3$

Model defined by:

- $\rho_F(S, w) = H$,
- $u = \gamma_{\Omega}(B)$,
- $\phi(w, 0) = ||w - c|| - r$

with $B = 1_{\rho_F(S, w) > 0}$ and $\gamma_{\Omega}(B) = DT(B) + DT(1 - B) + 0.5B$

- $\rho_F$ lacks the geometric term
  (not smooth except close to the target points)
- $\gamma_{\Omega}$ used as velocity like in $\mathcal{M}_1$
Motion model $\mathcal{M}_4$

Model defined by:

- $\rho_F(S, w) = H$,
- $u = \rho_F(S, w)$,
- $\phi(w, 0) = \gamma_\Omega(B) - b$

with $B = 1 - \rho_F(S, w)$ and $\gamma_\Omega(B) = DT(B) + DT(1 - B) + 0.5B$

- based on $\mathcal{M}_2$
- $\rho_F$ lacks the geometric term
Implementation

- Evolution: WENO5 approach
- Time step: CLF conditions, Godunov scheme
- Periodic reinitialization ($\mathcal{M}_2$, $\mathcal{M}_3$)
- GPGPU implementation
  - not real-time yet
Results

Planes

• three mutually orthogonal
• BB size: 2×2×2 [m³]
• # points: ~10k
Results

Planes

- two parallel, one orthogonal
- BB size: $2 \times 2 \times 2$ [m$^3$]
- # points: ~10k
Results

Bullet
• sphere and cylinder
• BB size: 2×2×2 [m^3]
• # points: ~10k
Results

Sphere-cone

- sphere and cone
- BB size: $2 \times 2 \times 2 \text{ [m}^3\text{]}$
- # points: ~10k
Results

Energy vs iterations for the segmentation of the artificial scene

Accuracy results of the proposed models

<table>
<thead>
<tr>
<th>Model</th>
<th>Mean [m]</th>
<th>Max [m]</th>
<th>Mean</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_1$</td>
<td>0.007</td>
<td>0.035</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M_2$</td>
<td>0.004</td>
<td>0.09</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M_3$</td>
<td>.014</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M_4$</td>
<td>0.021</td>
<td>0.21</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Results

Sensitivity to noise

noise-free

\[ \sigma^2 = 0.1 \]
Results

Sensitivity to noise

$\sigma^2 = 0.5$

$\sigma^2 = 2.0$
Results

Real data:

- noisy PCL from rotating Laser Scanner
- successive PCLs aggregated using ICP algorithm

input PCL

geometric rec.

segmented steps

segmented hand-rails
Conclusions

- Segmentation and reconstruction of parts of the surface based on its (estimated) geometric features
- Based on level-set methods
  - distance function uses geometric features
- Four possible models with different characteristics
- Robust against noise
- Efficient, massively parallel implementation (CUDA)
  - not real-time (yet)
  - ~1 sec for the displayed examples
Thank you!