RANSAC for Aligned Planes with Application to Roof Plane Detection in Point Clouds

Steffen Goebbels and Regina Pohle-Fröhlich
(Steffen.Goebbels@hsnr.de, Regina.Pohle@hsnr.de)

Niederrhein University of Applied Sciences - Institute for Pattern Recognition, Faculty of Electrical Engineering and Computer Science

GRAPP 2020, February 2020
Outline

- Data driven city model generation based on airborne laser scanning point clouds
- RANSAC with support vector
Background

Model based LoD 2 city model of North Rhine-Westphalia:
- Standard roof types, complex roofs are missing (churches, castles, . . .)
- Wrong roof slopes because of unrecognized dormers

More precise data based roof reconstruction required for
- Texture mapping
- Training data for segmentation of building structures in oblique aerial images with deep learning
Different approaches to plane segmentation in sparse point clouds

• RANSAC-Algorithmus
  • robust against noise
  • detects spurious planes (but local features like normals can be integrated)

• Hough-Transformation\(^1\)
  • difficult to avoid spurious planes
  • study shows that RANSAC is better suited for roof plane detection

• Region Growing based on normals
  • normals are distorted at ridge lines and step edges and due to noise

\(^1\)F. Tarsha-Kurdi, T. Landes, P. Grussenmeyer: Hough-transform and extended RANSAC algorithms for automatic detection of 3D building roof planes from lidar data. ISPRS Archives XXXVI (W52), 2007, pp. 407–412

Goebbels, Pohle-Fröhlich: RANSAC for Roof Planes. -3-
Our algorithm

- Detection of planar roof facets
- Reconstruction of roof polygons
  - Region growing of detected plane segments under consideration of ridge and step edges
  - Detection and simplification of facet boundaries and roof topology
  - Establishing orthogonality of edges by solving a mixed integer linear program
- Avoiding step edges by height adjustments
- Correcting planarity with a linear program
- Model simplification and CityGML generation
Computation of height map

Linear, triangulation based interpolation method
Segmentation of areas with homogeneous gradient direction prior to RANSAC

- Compute gradients of height map
- Exclude regions with small gradient lengths: flat roofs
- Classify angles of gradients
- Find connected areas for each angle class
- Estimate plane equations using RANSAC per area
Comparison with data-driven city model of North Rhine-Westphalia

Goebbels, Pohle-Fröhlich: RANSAC for Roof Planes. -7-
Comparison with data-driven city model of North Rhine-Westphalia (2)
However,...
Outline

- Data driven city model generation based on airborne laser scanning point clouds
- **RANSAC with support vector**
Set $G$ of footprint directions

Footprints can be taken from cadastral data or from point clouds.

Walls are visible as dense lines in photogrammetric or laser scanning point clouds.
RANSAC

procedure RANSAC($P$, $G$, iteration count $i$, threshold $\delta$)

$l_{\text{best}} := \emptyset$, $k = 1$

while $(k \leq i) \land (|l_{\text{best}}| < |P|)$ do

randomly select $\vec{p}_1, \vec{p}_2, \vec{p}_3 \in P$ with det[$\vec{p}_1, \vec{p}_2, \vec{p}_3] \neq 0$

$(\vec{n}, \rho) := \text{getPlaneParms}(\vec{p}_1, \vec{p}_2, \vec{p}_3, G)$

if $\vec{n}.z \not\approx 0$ then \hfill ▷ no wall

\[ l := \text{getInliers}(\vec{n}, \rho, P, \delta) \]

if $|l| > |l_{\text{best}}|$ then $l_{\text{best}} := l$, $\vec{n}_{\text{best}} := \vec{n}$, $\rho_{\text{best}} := \rho$

$k := k + 1$

if $|l_{\text{best}}| > 2$ then return $(\vec{n}_{\text{best}}, \rho_{\text{best}}, l_{\text{best}})$

else return “no plane”
Adjusted computation of plane parameters (1)

**procedure** \( \text{getPlaneParms}(\vec{p}_1, \vec{p}_2, \vec{p}_3, G) \)

\[
\vec{n}_0 := \frac{(\vec{p}_2 - \vec{p}_1) \times (\vec{p}_3 - \vec{p}_1)}{\| (\vec{p}_2 - \vec{p}_1) \times (\vec{p}_3 - \vec{p}_1) \|_2}
\]

\[
\rho_0 := < \vec{n}_0, \vec{p}_1 >
\]

**if** \( \vec{n}_0.z \approx 0 \) **then return** \((\vec{n}_0, \rho_0)\)

\[
\vec{n}_1 := \text{sign}(\vec{n}_0.z) \cdot \vec{n}_0
\]

\[
\rho_1 := \text{sign}(\vec{n}_0.z) \cdot \rho_0
\]

**if** \( \vec{n}_1.z \approx 1 \) **then return** \(((0, 0, 1), \rho_1)\)

\[
\vec{n}_2 := (\vec{n}_1.x, \vec{n}_1.y)/\| (\vec{n}_1.x, \vec{n}_1.y) \|_2.
\]

Normal \( \vec{n}_1 \) points upwards. Its projection to groundplane is \( \vec{n}_2 \).
Adjusted computation of plane parameters (2)

\[ m := -1 \]

\begin{align*}
\text{for } \vec{g} \in G & \text{ do} \\
& c := \vec{n}_2.x \cdot \vec{g}.x + \vec{n}_2.y \cdot \vec{g}.y \\
& \text{if } |c| > \cos(\alpha) \land |c| > m \text{ then} \\
& \quad \vec{h} := \text{sign}(c) \cdot \vec{g} \\
& \quad m := |c| \\
& d := \vec{n}_2.x \cdot (-\vec{g}.y) + \vec{n}_2.y \cdot \vec{g}.x \\
& \text{if } |d| > \cos(\alpha) \land |d| > m \text{ then} \\
& \quad \vec{h} := \text{sign}(d) \cdot \vec{g} \\
& \quad m := |d| \\
& \text{if } m = -1 \text{ then return } (\vec{n}_1, \rho_1)
\end{align*}

If this point is reached, the normal can be adjusted to a footprint direction.
Adjusted computation of plane parameters (3)

Adjust normal to footprint vector:

\[ \vec{r}_1 := \vec{p}_1 - \vec{p}_2, \quad \vec{r}_2 := \vec{p}_2 - \vec{p}_3, \quad \vec{r}_3 := \vec{p}_1 - \vec{p}_3 \]

for \( i \in \{1, 2, 3\} \) do

\[ a_i := \vec{r}_i \cdot \vec{h} \]
\[ b_i := |a_i| / \sqrt{\vec{r}_i \cdot \vec{r}_i} \]

for \( i \in \{1, 2, 3\} \) do

if \( b_i = \max\{b_1, b_2, b_3\} \) then

\[ a := a_i, \quad \vec{r} := \vec{r}_i, \quad \vec{p} := \vec{p}_i \]

if \( \vec{r} \cdot \vec{z} = 0 \) then \( l := 0 \);
else

\[ l := \frac{|\vec{r} \cdot \vec{z}|}{\sqrt{a^2 + \vec{r} \cdot \vec{z}}^2} \]

\[ \vec{n} := (l \cdot \vec{h} \cdot \vec{x}, l \cdot \vec{h} \cdot \vec{y}, \sqrt{1 - l^2}) \]

return \((\vec{n}, \langle \vec{p}, \vec{n} \rangle)\)
Minimum angle between initial normal and footprint directions

![Graph showing the minimum angle between initial normal and footprint directions.](image)
Sufficient number of iterations

If one selects a footprint direction from height map gradients, RANSAC only has to consider two instead of three different points in each iteration.

A plane with \(|I|\) inlier points in a pointcloud with \(|P|\) points is given. Then only

\[
\frac{\ln(1 - p)}{\ln \left(1 - \frac{|I|}{|P|} \cdot \frac{|I| - 1}{|P| - 1}\right)} < \frac{\ln(1 - p)}{\ln \left(1 - \frac{|I|}{|P|} \cdot \frac{|I| - 1}{|P| - 1} \cdot \frac{|I| - 2}{|P| - 2}\right)}
\]

iterations are required to find two different inliers of this plane with probability \(p\).
Slope adjustment with PCA

Using a Principal Component Analysis, the plane model can be optimized such that distances between plane and inliers become minimal subject to keeping the direction of the normal in the projection to the ground plane.
Angle changes due to PCA
Result based on standard RANSAC
... and based on modified RANSAC
Prof. Dr. Regina Pohle-Fröhlich Niederrhein University of Applied Sciences, iPatter Institute
E-Mail: regina.pohle@hsnr.de
Tel.: +49 2151 822 4760

Prof. Dr. Steffen Goebbels
Niederrhein University of Applied Sciences, iPatter Institute
E-Mail: steffen.goebbels@hsnr.de
Tel.: +49 2151 822 4648