Beautification of City Models based on Mixed Integer Linear Programming

Steffen Goebbels and Regina Pohle-Fröhlich

This is a self-archived version of a paper that appeared in Proceedings Operations Research 2018, Springer, Brüssel, 2019, pp. 119–125. The published article is available at https://link.springer.com/chapter/10.1007/978-3-030-18500-8_16

Abstract 3D City Models often are generated from sparse airborne laser scanning point clouds. Data-driven algorithms fit plane segments with the points and combine segments to watertight roof models. But low resolution laser scanning data lead to noisy boundary structures that have to be straightened. We propose a mixed integer linear program that rectifies directions of boundary edges according to the cadastral footprint and favors orthogonality. We apply this procedure to all connected components of the planar graph that represents polygonal boundaries of roof facets. The proposed method is suitable for the generation of large scale city models.

1 Introduction

Most algorithms that generate 3D building models either follow a data-driven or a model-driven methodology or combine both methodologies, see [10]. In a model-driven approach, parameterized standard roofs from a catalogue are fitted to point clouds obtained by airborne laser scanning or photogrammetry. Data-driven algorithms estimate plane segments and combine them to watertight roofs (cf. [5, 6, 7, 9]). However, publicly available point clouds often are sparse and contain less than ten points per square meter. This makes it difficult to estimate roof polygons exactly. In our data-driven framework, we use a Ramer Douglas Peucker algorithm to obtain straight edges, see [2, 4]. We also estimate rectangular structures and snap vertices to intersection lines between estimated planes (ridge lines). Nevertheless, the models still need improvement, especially along step edges, see Figure 2. Such model beautification also is useful in connection with reverse engineering of scanned 3D objects to CAD models, see [1] and the literature cited there.

The idea of this paper is to cautiously change the positions of vertices to obtain a maximum number of edges that are orthogonal to edges of the cadastral footprint.

Steffen Goebbels · Regina Pohle-Fröhlich
iPattern Institute, Niederrhein University of Applied Sciences, 47805 Krefeld, Germany
e-mail: {steffen.goebbels,regina.pohle}@hs-niederrhein.de

A changed vertex position has impact on at least two edges and the angles between them and their neighbor edges. Thus a global optimization problem arises. A similar problem is addressed in [11] and solved by minimizing a non-linear energy function through graph reduction techniques. However, amongst others, the condition of orthogonality is linear. Linear programming has already been successfully used to planarize roof polygons of city models, see [3]. For model beautification, binary variables are required to select the vertices that have to be changed. Thus, we introduce a mixed integer linear program (MIP) in the next section. The last section summarizes results.

2 A Linear Program for Beautification

Let D be a set of normalized direction vectors. Each vector of D is parallel or orthogonal to a significant edge of the building's cadastral footprint. We regard an edge as significant if it is longer than 2 m. We only consider the longest cadastral edges and limit D to eight elements. Each vector $d \in D$ is represented as a pair $(d.x, d.y)$, $\sqrt{(d.x)^2 + (d.y)^2} = 1$.

All building edges are projected to the x-y-ground plane. To reduce the number of edges and computational complexity, we first collapse sequences of edges to one edge if they have nearly the same orientation, i.e. angles between edges of the sequence are near π, and if the surrogate edge does not differ more than a threshold value from the replaced edges. The outcome is a planar graph. We determine all connected components of this graph because coordinate changes of a vertex do only have impact on the connected component of the vertex. For example, dormers in the roof's interior might lead to separate connected components.

Let E be the set of all non-trivial 2D edges $e = (e_1, e_2) = ((e_1.x, e_1.y), (e_2.x, e_2.y))$ of a connected component that fulfill following two conditions (see Figure 1): Each edge has to possess at least one vertex e_1 or e_2 that does not coincide with a vertex of the outer or inner cadastral footprint polygons. Also, an edge must not be completely covered by a footprint edge. We split $E = E' \cup E''$ into two disjoint sets. Edges in E' are allowed to change their orientation, whereas edges in E'' have to keep their original direction. We put all edges into E'' that are originally orthogonal to a vector of D. By keeping their orientation, we will reduce the number of binary variables.

V denotes the set of vertices belonging to edges in E, $V' \subset V$ is the set of vertices for which we allow position changes. Vertices in V' must not be used in footprint polygons and they must not represent intersection points of estimated roof's ridge lines. There might be vertices in V' that are not within the interior of the footprint but are positioned on a footprint edge. Let $V'' \subset V'$ be the set of these vertices. Our MIP changes their positions as well as positions of interior vertices. To this end we need float variables x_v^+, x_v^-, and y_v^+, y_v^- for each $v \in V$ that represent non-negative distances. For $v \in V \setminus V'$ their values are fixed set to zero.

We introduce binary variables $x_{e,d}$ which indicate that an edge $e \in E'$ becomes orthogonal to a direction vector $d \in D$:

$$x_{e,d} = \left\{ \begin{array}{l} 1 : e \text{ is transformed to become orthogonal to } d \\ 0 : \text{else} \end{array} \right\} \text{ for } e \in E', d \in D.$$

With a threshold value ε that limits coordinate changes, the MIP optimizes an

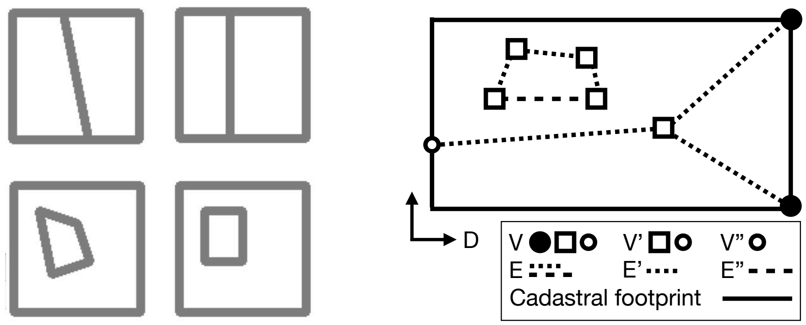

Fig. 1 Left: Two examples with a square as building footprint. The left sketches show original layouts, the right ones show the outcomes of optimization. The upper example has an edge with two vertices that are placed on footprint edges. The vertices have to stay on these edges. The vertices of the interior polygon in the lower example can be moved without such restrictions. Right: A roof's graph with two connected components is shown to illustrate the sets of edges $E = E' \cup E''$, the sets of vertices $V'' \subset V' \subset V$, and the set of footprint directions D.

objective function with the primary goal to find a maximum number of orthogonal edges:

$$\text{maximize } \left(\sum_{e \in E'} \sum_{d \in D} x_{e,d} \right) - \frac{1}{8 \cdot |V'| \cdot \varepsilon} \sum_{v \in V'} \left(x_v^+ + x_v^- + y_v^+ + y_v^- \right).$$

The secondary aim is to minimize coordinate changes. The factor $\frac{1}{8 \cdot |V'| \cdot \varepsilon}$, in which $|V'|$ is the number of elements of V', ensures that all possible coordinate changes contribute significantly less than one binary variable. We maximize the objective function subject to a list of restrictions:

Modified edges $e \in E'$ have to be orthogonal to direction $d \in D$ if $x_{e,d} = 1$:

$$\begin{aligned} -M \cdot (1 - x_{e,d}) \leq {} & (e_2.x + x_{e_2}^+ - x_{e_2}^- - e_1.x - x_{e_1}^+ + x_{e_1}^-) \cdot d.x \\ & + (e_2.y + y_{e_2}^+ - y_{e_2}^- - e_1.y - y_{e_1}^+ + y_{e_1}^-) \cdot d.y \leq M \cdot (1 - x_{e,d}), \end{aligned}$$

here M is chosen as a positive number that is larger than the longest occurring edge.

Thus, we implement the two conditions

$$
\begin{aligned}
&(x_{e_2}^+ - x_{e_2}^- - x_{e_1}^+ + x_{e_1}^-)\cdot d.x + (y_{e_2}^+ - y_{e_2}^- - y_{e_1}^+ + y_{e_1}^-)\cdot d.y + M\cdot x_{e,d}\\
&\le M + (e_1.x - e_2.x)\cdot d.x + (e_1.y - e_2.y)\cdot d.y,\\
&(-x_{e_2}^+ + x_{e_2}^- + x_{e_1}^+ - x_{e_1}^-)\cdot d.x + (-y_{e_2}^+ + y_{e_2}^- + y_{e_1}^+ - y_{e_1}^-)\cdot d.y + M\cdot x_{e,d}\\
&\le M + (-e_1.x + e_2.x)\cdot d.x + (-e_1.y + e_2.y)\cdot d.y.
\end{aligned}
$$

We do not change vertices on the cadastral footprint and intersection points of ridge lines. This gives the next set of restrictions:

$$x_v^+ = x_v^- = y_v^+ = y_v^- = 0 \text{ for all } v \in V \setminus V'.$$

Other vertices can be moved but only within the threshold distance $\varepsilon > 0$:

$$0 \le x_v^+, x_v^-, y_v^+, y_v^- < \varepsilon \text{ for all } v \in V'.$$

This condition might be changed to an adaptive one based on edge length.

Vertices in V'' are only allowed to change their positions so that they stay on the one footprint edge that they are positioned on, see upper example on the left side of Figure 1. Thus, for all $v \in V''$ we introduce a parameter variable $0 \le r_v \le 1$ and require

$$
\begin{aligned}
x_v^+ - x_v^- - (b_v.x - a_v.x)\cdot r_v &= a_v.x - v.x\\
y_v^+ - y_v^- - (b_v.y - a_v.y)\cdot r_v &= a_v.y - v.y,
\end{aligned}
$$

where a_v and b_v denote the 2D vertices of the corresponding footprint edge.

To reduce the degrees of freedom, we set $x_{e,d} = 0$ for all $e \in E'$, $d \in D$ with

$$\frac{|(e_2.x - e_1.x)\cdot d.x + (e_2.y - e_1.y)\cdot d.y|}{\sqrt{(e_2.x - e_1.x)^2 + (e_2.y - e_1.y)^2}} > \left|\cos\left(\frac{\pi}{2} + \alpha\right)\right|$$

such that for each model edge we only consider roughly orthogonal cadastral footprint directions. Maximum deviation from $\pm\frac{\pi}{2}$ is determined by the threshold angle α. Another set of restrictions keep the orientation of edges that are orthogonal to a footprint direction $d \in D$ from the beginning: For all $e \in E''$ let

$$(x_{e_2}^+ - x_{e_2}^- - x_{e_1}^+ + x_{e_1}^-)\cdot(e_1.y - e_2.y) + (y_{e_2}^+ - y_{e_2}^- - y_{e_1}^+ + y_{e_1}^-)\cdot(e_2.x - e_1.x) = 0.$$

Finally, feasible solutions fulfil $\sum_{d\in D} x_{e,d} \le 1$ for all $e \in E'$.

We follow an optimistic approach and, for performance reasons, do not check if edges cross other edges due to position changes. In that case, self intersections of polygons or intersections between different connected components occur. Our existing framework for building model generation resolves such situations by cutting polygons into pieces. If a problem cannot be resolved then the threshold value ε will be reduced and the model will be re-computed. The algorithm starts with a given threshold value ε and then iteratively divides the threshold value by two. However, we only update vertices with their optimized positions if they do not leave the area of the cadastral footprint.

3 Results

Our implementation is based on the GNU Linear Programming Kit library GLPK [8]. We apply the MIP to city model generation of the square kilometer with 1829 buildings of Krefeld that covers the building of our institute. This tile is visualized in Figure 2. Table 1 summarizes resulting sizes of non-trivial problem instances with at least one changeable edge, i.e. one binary variable. The improved city model is visualized in Figure 2.

Table 1 Results for a city model of a square kilometer (UTM intervals $[32330000, 32331000] \times [5687000, 5688000]$: For parameter combination $\varepsilon = 1$ m, $\alpha = \frac{\pi}{6}$, there exist 1826 non-trivial problem instances for connected components, 61 reach the time limit of two seconds (not included in running time data, a larger time limit does not increase the number of successful instances significantly). There are 1876 instances for parameters $\varepsilon = 2$ m, $\alpha = \frac{\pi}{4}$, of which 89 exceed the time limit. Running times are measured on one kernel of a Macbook Pro (2013) with 2.4 GHz i5 processor.

		minimum	maximum	arithmetic mean	median	quartiles
$\varepsilon = 1$ m, $\alpha = \frac{\pi}{6}$	variables	9	2709	66.69	35	20, 45
	binary variables	1	1210	10.34	4	2, 8
	conditions	2	3004	43.61	20	11, 39
	changed vertices	0	233	4.01	3	2, 5
	running time [ms]	0.09	1697.96	17.45	0.46	0.29, 1.28
$\varepsilon = 2$ m, $\alpha = \frac{\pi}{4}$	variables	9	2951	68.05	36	20, 45
	binary variables	1	1500	12.51	4	2, 9
	conditions	2	3679	48.88	22	11, 42
	changed vertices	0	261	4.44	3	2, 5
	running time [ms]	0.12	1765.27	19.08	0.52	0.31, 1.51

An artifact of optimization with an l_1-norm instead of using least squares is that larger individual coordinate changes might occur. For example, in Figure 1 the vertical line in the upper left sketch is not optimized to be in the middle of the square.

References

1. Gauthier, S., Puech, W., Bénière, R., Subsol, G.: Orientation beautification of reverse engineered models. In: Proceedings International Conference on Computer Graphics Theory and Applications (GRAPP 2018), pp. 91–100. SCITEPRESS, Funchal (2018)
2. Goebbels, S., Pohle-Fröhlich, R.: Quality enhancement techniques for building models derived from sparse point clouds. In: Proceedings International Conference on Computer Graphics Theory and Applications (GRAPP 2017), pp. 93–104. SCITEPRESS, Porto (2017)
3. Goebbels, S., Pohle-Fröhlich, R., Rethmann, J.: Planarization of CityGML models using a linear program. In: Operations Research Proceedings (OR 2016 Hamburg), pp. 591–597. Springer, Berlin (2016)
4. Goebbels, S., Pohle-Fröhlich, R.: Roof reconstruction from airborne laser scanning data based on image processing methods. ISPRS Ann. Photogramm. Remote Sens. and Spatial Inf. Sci. **III-3**, 407–414 (2016)

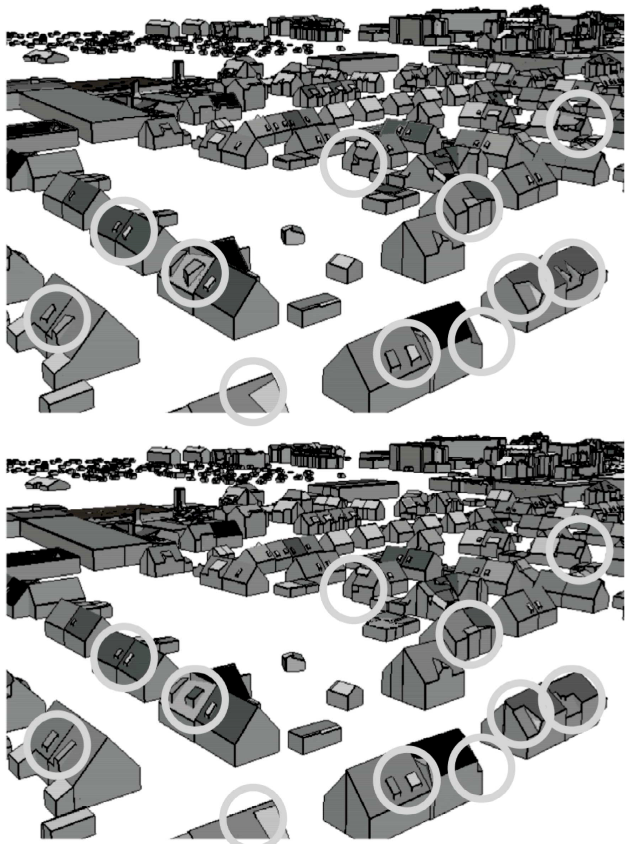

Fig. 2 City model before and after beautification: Some changes are marked with circles. Roofs and walls are simplified differently due to post-processing.

5. Haala, N., Kada, M.: An update on automatic 3D building reconstruction. ISPRS Journal of Photogrammetry and Remote Sensing **65**, 570–580 (2010)
6. He, Y.: Automated 3D Building Modeling from Airborne LiDAR Data (PhD thesis). University of Melbourne, Melbourne (2015)
7. Henn, A., Gröger, G., Stroh, V., Plümer, L.: Model driven reconstruction of roofs from sparse lidar point clouds. ISPRS Journal of Photogrammetry and Remote Sensing **76**, 17–29 (2013)
8. Makhorin, A.: The GNU Linear Programming Kit (GLPK). Free Software Foundation, Boston, MA (2009)
9. Perera, S.N., Maas, N.G.: A topology based approach for the generation and regularization of roof outlines in airborne laser scanning data. DGPF Tagungsband **21**, 1–10 (2012)
10. Wang, R., Peethambaran, J., Dong, C.: Lidar point clouds to 3D urban models: A review. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **11**(2), 606–627 (2018)
11. Yan, J., Zhang, K., Zhang, C., Chen, S.C.: Automatic construction of 3-D building model from airborne lidar data through 2-D snake algorithm. IEEE Transactions on Geoscience and Remote Sensing **53**(1), 3–14 (2015)