Building Footprint Simplification Based on Hough Transform and Least Squares Adjustment

Richard Guercke, Monika Sester
Institut für Kartographie und Geoinformatik, Leibniz Universität Hannover, Appelstraße 9a, 30167 Hannover

Abstract. The approach described in this paper can be used to straighten out jagged building footprint shapes and for the simplification of detailed building footprints. In a first step, the outline of the polygon is sampled into small line segments. These line segments are transformed into Hough space. The lines corresponding to the peaks in the Hough buffer are used to generate initial hypotheses for the segments forming the outline of the simplified polygon. These line hypotheses are refined by a least squares adjustment procedure in which the distance to the segments of the original outline is minimized while constraints like parallel or perpendicular segments can be enforced. In a final step, the segments are linked to form a closed polygon. First tests show that the approach produces a small number of line segments that usually approximate the original polygon very well.

Keywords. Generalization, Hough Transform, Least Squares, Building Footprint, LOD 1

Introduction and Related Work

For a wide range of applications, it can be necessary to reduce the complexity of building footprint polygons. For the tests we conducted, we had two generic scenarios in mind: The simplification of highly detailed cadastre data for small scale maps, or applications (e.g. display in mobile devices) that do not need and/or cannot cope with the fine granularity of the cadastre data – especially in the approximation of arcs; the second scenario is to extract plausible building footprint shapes from jagged polygons generated through LIDAR data analysis., These scenarios differ in the fact that the latter contains a lot of redundant points, whereas the former is already a minimal representation.

General methods for line simplification (e.g. Douglas-Peucker) cannot be applied, as they do not take the characteristics of the buildings into account, namely that the walls of buildings are often constructed at right angles or aligned with each other, and those relations should be retained in the generalization process to produce realistic generalized building shapes. To this end, rule based (Staufenbiel, 1973, Sester, 2005) or agent based methods have been developed (Lamy et al., 1999).

For the simplification of highly redundant data like building outlines derived from LIDAR data different methods have been proposed, e.g. Maas & Vosselman (1999) use geometric moments to determine the main orientation parameters for gable roof buildings and use the direction of the ridge line as an approximation for the main direction of the building (modulo 90 degrees). Shan & Sampath (2007) use straight lines in the main direction of the buildings to approximate the shape and least squares adjustment for the adaptation to the original boundary points.

The approach presented here is closely related to (Sester & Neidhart, 2008): In a first step, a set of initial line hypotheses is generated. While Sester and Neidhart (2008) use RANSAC to do this, the new approach uses Hough analysis. In a second step, the hypotheses are refined using a least squares adjustment process in which the segments are shifted to fit the original shape and to enforce relations between segments. In the new approach, various steps like transferring references to associated parts of the original outline between segment hypotheses are introduced to improve the flexibility of the approach. Additionally, the design matrix of the least squares iterations was changed slightly.

The approach presented in this paper is able to treat both kinds of input information. It has not been tested thoroughly yet, but the results it produced so far are promising. Before it can be considered for unsupervised productive use, however, some additional research into the refinement of the algorithms will be necessary.

1. Generating Line Hypotheses

In order to detect characteristic line segments that represent the main directions of the footprint polygon, we subject the footprint to a Hough transform-based analysis.

The outline of the polygon is first divided into small line segments (“snips”). The starting points of these snips are used to fill a Hough buffer; the black dots in Figure 1 show the starting points of the snips for a jagged footprint polygon extracted from LIDAR data. In the standard test case, the length of the snips was 0.1m.

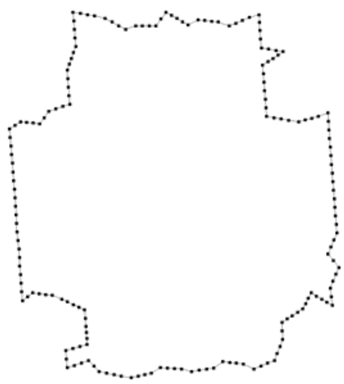


Figure 1: A jagged footprint polygon.

For the representation of the lines in Hough space we use the common form of orientation (angle φ) and distance d to the origin. Since we know the direction of the snips in the original polygon, we use a “full-angle” Hough buffer covering the full range of 360°. The angles are given in a geographic system with 0° facing in the positive y-direction (“North”) and clockwise orientation. Figure 2 shows a plot of the Hough buffer corresponding to the polygon in Figure 1; each pixel covers an angle range of $\Delta\varphi = 0.5°$ and a distance range $\Delta d = 0.25$m.

All possible lines through a given point form a (co-)sine wave in Hough space. Since we know the direction of the original polygon edge for each snip, we could represent each snip by a single point in Hough space. Unfortunately, the different line segments that contribute to a main direction ("stroke") of a jagged polygon can have very different orientations. For this reason, we define an angle tolerance ε_φ and add the sine wave corresponding to the starting point of the snip for the angles $[\varphi_0 - \varepsilon_\varphi, \varphi_0 + \varepsilon_\varphi]$ to the accumulator.

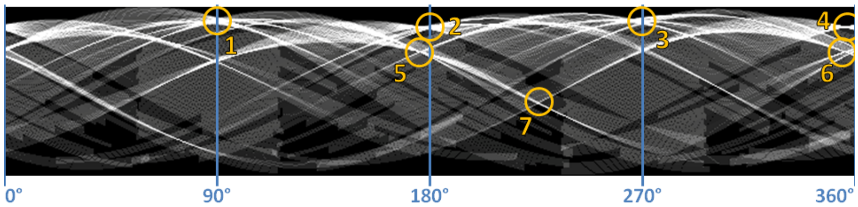


Figure 2: Hough buffer corresponding to the polygon in Figure 1.

In order to generate line segment hypotheses from the snips, the following algorithm is applied:

1. Render all snips in the snip pool to the Hough accumulator.
2. Select the center of the cell with the highest value as the current line *l*.
3. Collect all snips with a distance of less than ε_{line} (=0.8m in the standard test case) from *l* in a set *supp* (support).
4. Project all segments in *supp* on *l*; aggregate all snips with a projected distance of less than d_{chunks} into a line segment *s* and store all line segments with a length of more than l_{min}.
5. Remove the snips supporting the extracted line segments from the snip pool and store them with the line segments.
6. Repeat steps 1-5 until no new segments are detected.

In Figure 2, the peaks in the Hough buffer are displayed labeled by the order in which they were extracted. The buffer itself did, of course, look different after each iteration. Figure 3 shows the resulting line segments – the polygon is oriented clockwise so segment 1, for example, faces to the right ("east" corresponding to 90° in the Hough buffer). Line no. 6 is divided into the line segments *6a* and *6b* because there is a gap of more than d_{chunks} (=1.2m in the standard parameter set) in the support set.

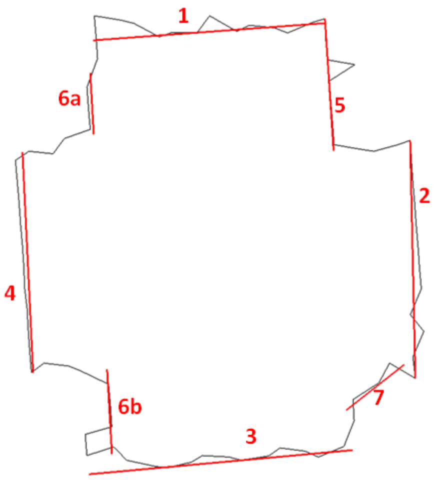


Figure 3: Initial line segments.

Problems can occur if prominent directions are formed by many short segments with gaps of more than d_{chunks}. In such a case, the algorithm in its current form would not detect all segments. Especially at smaller scales, this can become a serious problem that can be solved by detecting multiple peaks in the Hough buffer simultaneously and testing them all.

2. Adjustment

The goals of the refinement step are to align the line hypotheses more closely with the original polygon and to emphasize typical relationships between segments in building footprints, especially parallel and perpendicular segments. Additionally, the assignment of supporting snips to the different line segments is readjusted.

The first step (S1) in the adjustment loop is to determine the sequence in which the line segments form the outline of the new polygon. This is done by traversing the snips in the order in which they form the outline of the original polygon and enumerating the line segments to which the snips belong. A possible sequence for the polygon in Figure 3 is [1, 5, 2, 7, 3, 6b, 4, 6a]. Due to alternating assignments of snips to different segments, parts of the sequence may be repeated. Since we want to establish an ordering of the segments, such repetitions are not desirable and therefore removed from the sequence.

In the second step (S2), the snips assigned to all adjacent segments are investigated. If a snip assigned to one of the segments fits the other one better (has a smaller distance to the other segment), it is tentatively assigned to the other segment. Then the projected snips are investigated again and the segment is split if d_{chunks} is exceeded as in step 4 in section 1. The chunk with the greatest overlap with the original segment replaces the segment and the remaining snips are returned to the other segment. If a segment is too short after this process, it is removed and its remaining supporting snips are return to the snip pool. At this point, it should be tested if slightly offset parallel segments can be joined and if segments with a “stairs-shaped” support set should be split. This is, however, not implemented yet in the current prototype.

After that, the relationships between all segments are established in step S3: If two segments are almost perpendicular or parallel (within an angle threshold $\Delta\varphi_{rel}$), this relation is stored.

In step S5, a least squares adjustment iteration is performed that combines a fitting of the line segments to the original polygon and an emphasizing of the relationships between the segments. The variables of the adjustment are the x and y components a_i and b_i of the normal vector of each segment i and its distance d_i to the origin. The linearized observation equations are:

1) Normal vectors are normalized:
$$\underline{a_i} \cdot a_{i,old} + \underline{b_i} \cdot b_{i,old} = 1$$
for all segments i.
2) Fit the segments to the snips:
$$\underline{a_i} \cdot p_x + \underline{b_i} \cdot p_y + \underline{d_i} = 0$$
for all starting points p of a snip.
3) Shift the segments to fit the snips (without change of direction):
$$-\underline{d_i} = a_{i,old} \cdot p_x + b_{i,old} \cdot p_y$$
for all starting points p of a snip.
4) Enforce relations between segments:
   - For all perpendicular segments i and j:
   $$\overrightarrow{n_{0,i}} \cdot \overrightarrow{n_{0,j}} = a_i \cdot a_j + b_i \cdot b_j = 0 \rightarrow$$
   $$\underline{a_i} \cdot a_{j,old} + \underline{a_j} \cdot a_{i,old} + \underline{b_i} \cdot b_{j,old} + \underline{b_j} \cdot b_{i,old} = 0$$
   - For all parallel segments i and j:
   $$\left\| \overrightarrow{n_{0,i}} \times \overrightarrow{n_{0,j}} \right\| = a_i \cdot b_j - b_i \cdot a_j = 0 \rightarrow$$
   $$\underline{a_i} \cdot b_{j,old} + \underline{b_j} \cdot a_{i,old} - \underline{b_i} \cdot a_{j,old} + \underline{a_j} \cdot b_{i,old} = 0$$

The weights of the different types of observations are different: The normalization is necessary constraint and receives a weight of 10^8; the weight W_2 of the snip fitting equations of type is set to 1, the parallel shift equations of type 3 receive a relative weight W_3 that is increased with the number of iterations. The weight W_4 of the relations is also increased with the number of iterations; additionally, there is a further increase of weight for relations between adjacent line segments.

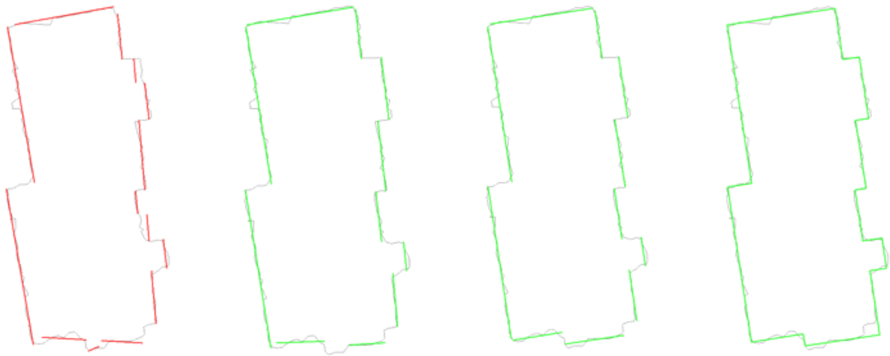

(a) (b) (c) (d)

Figure 4: Different adjustment steps for a footprint polygon: original segments (a), after 3 iterations (b), after 12 iterations (c), and final segments (d).

Only one (or a few) iteration of the linearized quadratic least squares adjustment in S5 is performed. After that, the previously unassigned snips are tested if they can be assigned to the shifted segments, and those snips that are outside the buffer around the segments after the adjustment are returned to the pool of unassigned snips. Then the whole adjustment loop starts again with step S1. Usually, 10 to 20 global iterations were sufficient.

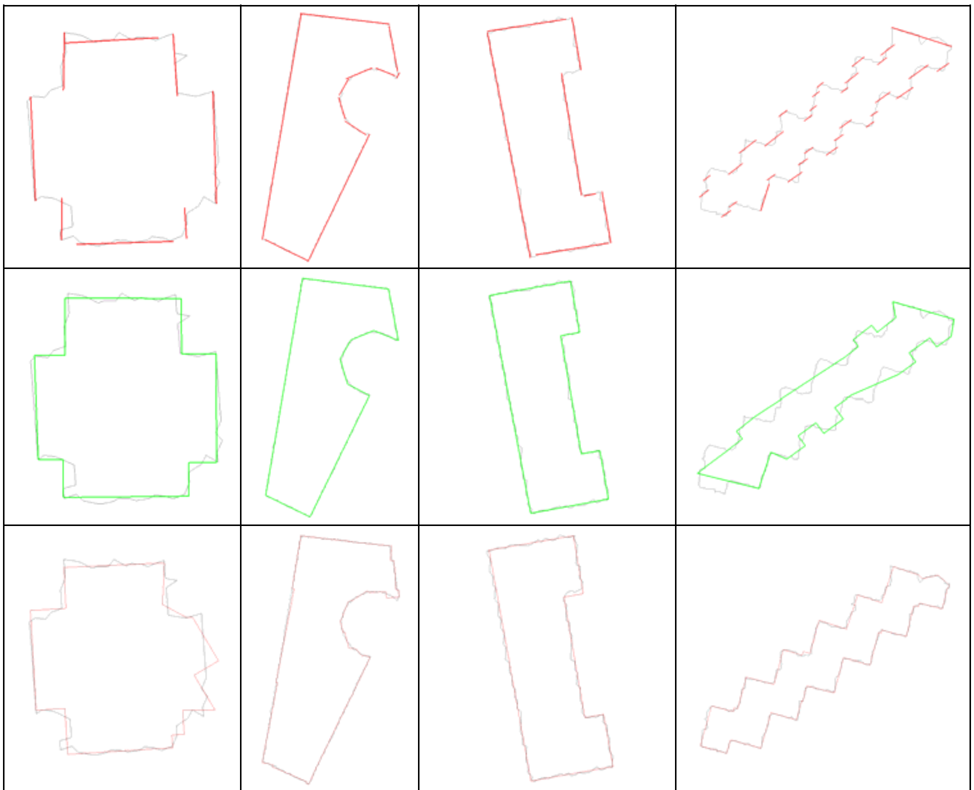

(a) (b) (c) (d)

Figure 5: Stitching the gaps between segments.

Finally, the line segments are "stitched" together as shown in Figure 5: Two adjacent segments are joined in their intersection point if this does not change the length of a segment by a greater amount than ε_{line} (a). If case (a) applies, cases (b)-(d) are not tested. Parallel segments are linked by a perpendicular segment halfway between the end of the first and the start of the second (b). For perpendicular segments, perpendicular segments starting at the end of the first and the start of the second (linked at their intersection) are inserted (c). In all other cases, a new segment linking the end of the first to the start of the second is inserted (d).

3. Results

We had the opportunity to compare the results of the new algorithm to those obtained using the approach presented in (Sester & Neidhart, 2008). These results are used as the reference in the assessment of the algorithm. In general, the new algorithm generated fewer and often more meaningful segments with comparable tolerance bands and similar overlap ratios with the original polygon. For moderately complex polygons like the one shown in Figure 6(a), the new approach usually produced 25% to 30% fewer segments than the reference approach without a relevant loss of quality, in the remarkable case shown in Figure 6(b), 75% of the segments are saved with increased geometric accuracy. In simple cases like the one illustrated Figure 6(c), there was often at least one small offset in one of the main segments (not noticeable in the figure) creating an additional segment in the reference data set.



#Segments		#Segments		#Segments		#Segments	
new	ref	new	ref	new	ref	new	ref
12	18	10	43	8	9	(30)	(44)
(a)		(b)		(c)		(d)	

Figure 6: Top row: original segments (new), middle row: final segments (new), bottom row: final segments (reference)

In special cases, especially with “stairs” patterns, the new algorithm has problems. Such a case is illustrated in Figure 6(d). These problems can probably be solved by a more detailed analysis of the segment hypotheses, by investigating multiple peaks in the initial Hough accumulator simultaneously, and by generating segments during the adjustment phase.

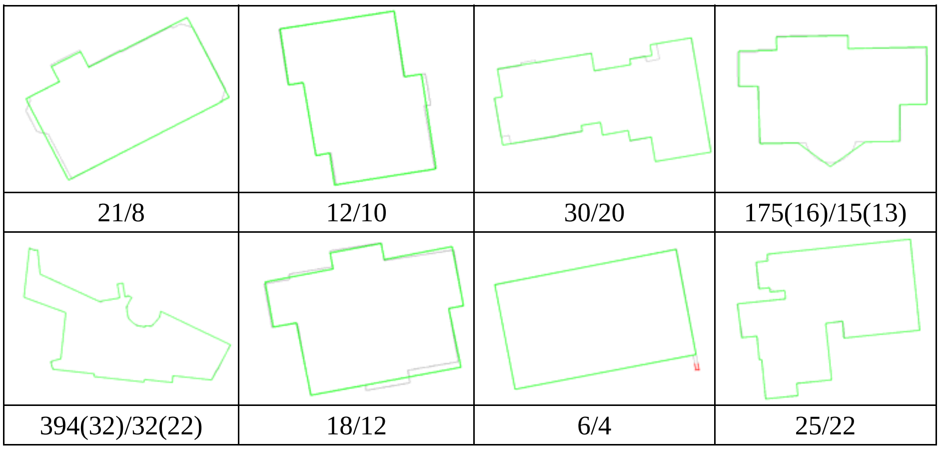


Figure 7: Simplification of cadastre building footprints with number of segments (original/simplified); numbers in brackets: without segments approximating arcs.

Figure 7 shows some examples illustrating the fact that the algorithm can be used to simplify building footprints from cadastre data sets. The parameter set was identical to the one used for straightening the jagged data set. Results of more detailed tests with parameters for different levels of simplification are going to be available for the presentation at the workshop.

The case with 6/4 segments shows that the approach does not guarantee a maximum Hausdorff distance between the original and the approximating polygon: The snips highlighted red in this drawing are, for example, outside a buffer of ε_{line} around the approximating polygon.

4. Conclusions and Outlook

If enough initial line segments can be extracted from the original polygon, the new approach usually produces simplified polygons with a small number of vertices that match the original shape nicely while retaining or emphasizing the relations between parallel and perpendicular edges.

Some situations have been identified and analyzed in which the approach does not provide the best possible results; especially a more thorough analysis of the individual line hypotheses (e.g. by a local Hough analysis) can provide additional insight into the structure of the original polygon and make it possible to use the algorithm as a basis for feature extraction procedures – for example, to detect and classify intrusions and additions which would help to develop semantics-based generalization operations.

While segments can be deleted in the adjustment process, it is also often desirable to generate new segments in the course of the adjustment. This can be done by subjecting the currently unassigned snips to a new Hough analysis. Additionally, it can be sensible to merge (almost) collinear adjacent segments.

Testing different hypotheses in the final process of stitching the gaps between the line segments and including the stitching segments in the adjustment process can also increase the quality of the result.

Further research is needed to determine sensible parameters for different resolutions and to deal with cases in which the first maxima in the Hough buffer do not produce enough segments of sufficient length, e.g. for jagged “stairs” patterns as shown in Figure 6(d).

References

Douglas, D. & Peucker, T., 1973, Algorithms for the reduction of the number of points required to represent a digitized line or its caricature, The Canadian Cartographer 10(2), 112-122.

Lamy, S., Ruas, A., Demazeau, Y., Jackson, M., Mackaness, W. & Weibel, R., 1999, The Application of Agents in Automated Map Generalization, in: Proceedings of the 19th International Cartographic Conference of the ICA, Ottawa, Canada.

Sester, M. and H. Neidhart, 2008: Reconstruction of Building Ground Plans from Laser Scanner Data, Proceedings of the AGILE, Girona, Spain, 2008.

Sester, M., 2005: Optimizing Approaches for Generalization and Data Abstraction, in: International Journal of Geographic Information Science, Vol 19, Nr. 8-9, pp. 871-897, 2005.

Shan, J. & A. Sampath, 2007: Urban Terrain and Building Extraction from Airborne LIDAR Data; in: Urban Remote Sensing (Q. Weng & D. A. Quattrochi, eds.), CRC Press January 2007.

Staufenbiel, W.,1973: Zur Automation der Generalisierung topographischer Karten mit besonderer Berücksichtigung großmaßstäbiger Gebäudedarstellungen, Doktorarbeit, Fachrichtung Vermessungswesen, Universität Hannover.