Building Footprint Regularization : From Vectorization to Deep Learning

This manuscript (permalink) was automatically generated from kshitijrajsharma/building-regularization-research@3bba092 on May 25, 2025.

Authors

- **Kshitij Raj Sharma**
  0000-0002-2123-3917 · kshitijrajsharma · @krschap@mastodon.social
  Department of Geoinformatics, Paris Lodron University, Salzburg, Austria

✉ — Correspondence possible via GitHub Issues

Abstract

Geographic information systems (GIS) and cartographic applications typically require building footprints as precise vector polygons, rather than raster masks [1]. Building footprint regularization refers to the process of refining raw building outlines (e.g. from remotely sensed imagery or LiDAR) into clean polygon shapes that conform to expected geometric constraints (such as orthogonal corners or aligned edges). The goal is to eliminate irregular artifacts (noisy jags, misalignments) while preserving the true shape, so that the footprints are cartographically suitable for maps In many cases, regularization assumes buildings are rectilinear structures with predominantly 90° corners : an assumption that, while not universally true, holds for most residential and industrial buildings. This review traces the evolution of footprint regularization methods from early vectorization algorithms in the 1990s through modern deep learning approaches in the 2020s.

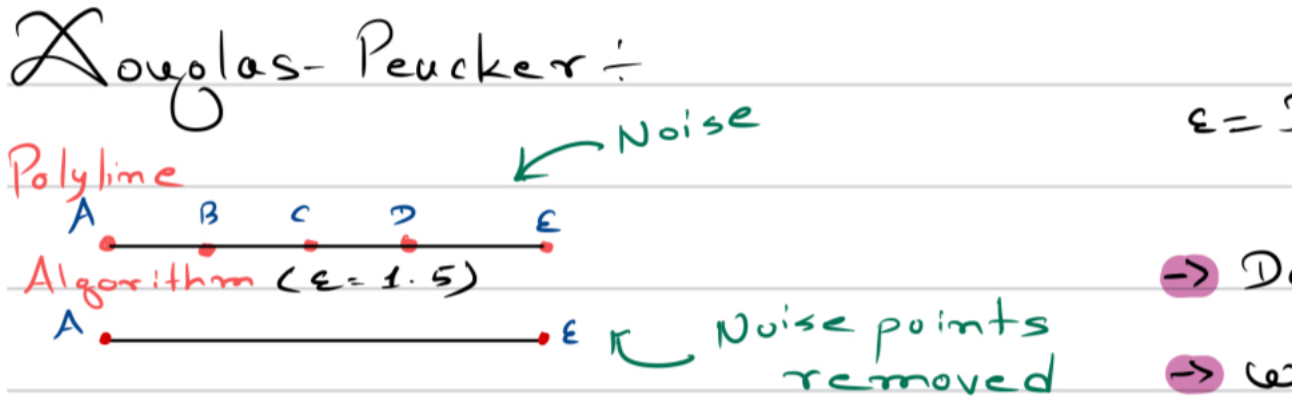
We focus on 2D footprint outline techniques (planimetric building outlines) and exclude full 3D building reconstruction or roof modeling. Key developments and representative methods are discussed for each era, highlighting their algorithms, use cases, strengths, and limitations. We then compare traditional versus deep learning-based methods in terms of performance, flexibility, accuracy, and integration into GIS workflows. The review draws on peer-reviewed research and real-world implementations (including open- source tools and commercial pipelines) to provide a comprehensive perspective for remote sensing and GIS professionals.

Geometric and Heuristic Methods (1990s - 2000s)

Edge Detection and Line Fitting: Early building extraction in the 1990s relied on low-level image processing and geometric heuristics. For example, Huertas and Nevatia (1988) developed a system to detect buildings in aerial images by finding rectangular clusters of edges (lines) and using shadow cues to distinguish buildings from other structures [2] . Building polygons often consist of jagged lines. Guercke and Sester [3] use Hough-Transformation to refine such polygons.

Those approach and similar ones could identify simple rectangular building footprints, but often produced polygons with jagged (bearing in mind they don't take into account the building shape itself rather the outline), noisy outlines. To clean such outlines, researchers applied line simplification algorithms from cartography, notably the Ramer–Douglas–Peucker algorithm : to remove small zig-zags and reduce vertex count while approximating the shape (which is still used to the date) [4/].

The Douglas–Peucker algorithm (originally from 1973) [5] became a common post-processing step to "compress" or simplify building polygon geometry.



Overall, early methods were largely rule-based: edges and corners were detected via image filters, and building shapes were assembled by connecting these primitives under geometric constraints defined by human experts.

Regularization via Hough Transform: By the 2000s, more sophisticated heuristics were introduced to enforce regularity in building outlines. A prominent tool was the Hough Transform for line detection. Hough transform is a feature extraction method used in image analysis. Hough transform can be used to isolate features of any regular curve like lines, circles, ellipses, etc. Hough transform in its simplest from can be used to detect straight lines in an image.[st1739/hough-transform-287b2dac0c70?] For instance, Guercke and Sester (2011) proposed a footprint simplification method that takes an initial digitized outline (which might be jagged) and uses a Hough Transform to identify the dominant line orientations; close-to-collinear segments are merged and adjusted by least-squares to align with those dominant directions [7].

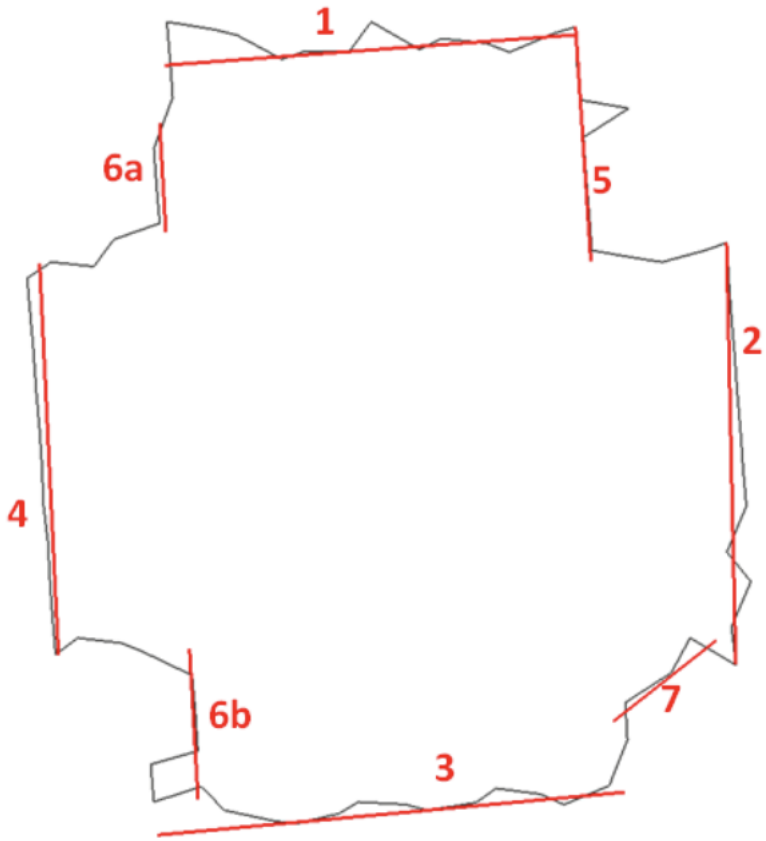


Figure 1: Initial hough transofrmation line segment explained by Guercke and Sester (2011)

The result is a cleaner, rectilinear footprint where spurious bends are straightened and most angles are ~90° or 180° [8]. Shiyong Cui et al. (2012) similarly applied the Hough transform to grouping line segments into two perpendicular families corresponding to a building's principal directions . They constructed an initial graph of line segments, pruned edges that lacked image contrast (assuming they were false boundaries), and then detected closed cycles in the graph to form building polygons [8].

This yielded neatly rectangular footprints for buildings aligned to the two main axes, although the method was inherently limited to rectilinear structures. Tian and Reinartz (2013) extended the idea to allow two arbitrary dominant orientations (not necessarily parallel/perpendicular to the image axes), enabling footprints with an oblique alignment (e.g. buildings rotated on the ground) [8].

These Hough-based methods exemplify how prior knowledge of building shape (e.g. most buildings have parallel opposite walls and right-angle corners) was hard-coded into algorithms well before machine learning became common. The advantage was that the output polygons were regular by design : straight lines, right or consistent angles; making them immediately usable for mapping. However, the success of these methods depended on reliable low-level edge detection. In practice, missing or spurious line segments could cause incomplete or incorrect polygons. Methods like Cui's required a clear dominance of two perpendicular directions; complex or curved buildings, or those with more than two prevailing orientations, fell outside their scope. Hough transform is considered as a computational complex in terms of algorithm itself & often require postprocessing techniques like snapping/merging lines or form cycles to create valid polygons[st1739/hough-transform-287b2dac0c70?]

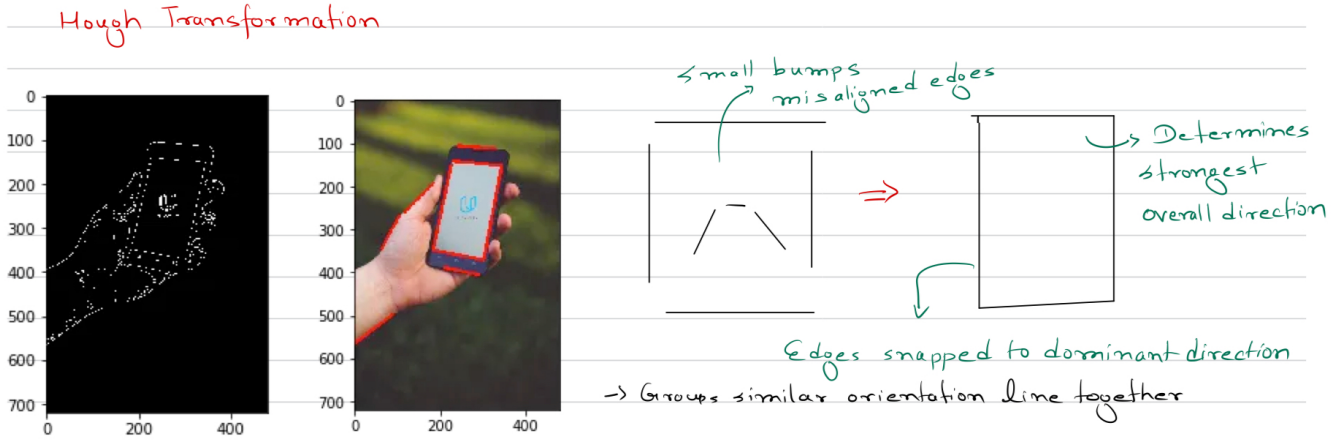


Figure 2: A simple Hough transformation explaination

References

1. **PolyWorld: Polygonal Building Extraction With Graph Neural Networks in Satellite Images**
   Stefano Zorzi, Shabab Bazrafkan, Stefan Habenschuss, Friedrich Fraundorfer
   (2022)
   https://openaccess.thecvf.com/content/CVPR2022/html/Zorzi_PolyWorld_Polygonal_Building_Extraction_With_Graph_Neural_Networks_in_Satellite_CVPR_2022_paper.html

2. **Automatic Building Footprint Extraction from Multi-Resolution Remote Sensing Images Using a Hybrid FCN**
   Philipp Schuegraf, Ksenia Bittner
   ISPRS International Journal of Geo-Information (2019-04-12) https://doi.org/ggwr8s
   DOI: 10.3390/ijgi8040191

3. **Building Footprint Simplification Based on Hough Transform and Least Squares Adjustment**
   Richard Guercke, Monika Sester
   Proceedings of the 14th Workshop of the ICA Commission on Generalisation and Multiple Representation (2011-07)

4. **Automated Building Footprint Extraction (Part 3): Model Architectures • Element 84** (2022-11-09) https://element84.com/software-engineering/automated-building-footprint-extraction-part-3-model-architectures/

5. **Algorithms for the reduction of the number of points required to represent a digitized line or its caricature**
   David H Douglas, Thomas K Peucker
   The Canadian Cartographer (1973)

6. **Medium: Read and write stories.**
   Medium
   https://medium.com

7. https://www.mdpi.com/2220-9964/8/4/191

8. **Rectilinear Building Footprint Regularization Using Deep Learning**
   Philipp Schuegraf, Zhixin Li, Jiaojiao Tian, Jie Shan, Ksenia Bittner
   ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences (2024-06-10) https://doi.org/g9k82r
   DOI: 10.5194/isprs-annals-x-2-2024-217-2024