Accumulator-Based Three Dimensional Building Boundary Reconstruction framework for Satellite Images

by

Willard Tapiwa Mapurisa

In fulfilment of the requirements for the degree of

Doctor of Philosophy in Engineering

Department of Geomatics

Faculty of Engineering and Built Environment

UNIVERSITY OF CAPE TOWN

Submitted: March 11, 2024

Supervisor: Dr George Sithole

Administrative Supervisor: Dr Philippa Tumubweinee

DECLARATION

I, Willard Tapiwa Mapurisa , hereby declare that the work on which this dissertation/thesis is based is my original work (except where acknowledgements indicate otherwise) and that neither the whole work nor any part of it has been, is being, or is to be submitted for another degree in this or any other university. I empower the university to reproduce for the purpose of research either the whole or any portion of the contents in any manner whatsoever.

Signature: _____ Signed by candidate

Date: March 11, 2024

Acknowledgements

I would like to thank Dr George Sithole for guidance, support and opportunities he has given me from the beginning of my postgraduate journey. His patience and efforts to ensure the completion of this work are greatly appreciated.

I would like to thank Luxcarta for providing the data used in this investigation. Their work in Three Dimensional Mapping of our environment inspired this research.

I would like to thank my wife, Dr Bolelang Sibolla and kids for the support and encouragement. Your unwavering support throughout this journey is highly appreciated.

Finally, I would like to thank God for making all things possible.

ABSTRACT

Three Dimensional (3D) building boundary detection and reconstruction from satellite images has various applications in computer vision such as 3D city modelling, radio network planning, urban planning, mapping, and navigation. Traditionally, 3D building models have been produced from multiview satellite images using dense pixel matching. The point clouds or Digital Surface Model (DSM)s produced from the matching are then used to estimate 3D building boundaries. From these point clouds 3D wireframe models can be generated by recovering the surface boundaries of objects in the point cloud. Ideally, it is desirable for recovered edges to follow the object edges without generalisation and with good localisation. However, the accuracy and fidelity of most of the reconstructed boundaries are diminished because of point mismatches, occlusions, and the failure to recover sharp object discontinuities during image matching. An alternative 3D boundary detection and reconstruction approach that has been less widely explored is the use of edge features exclusively for 3D reconstruction. The advantage presented by using edge features in 3D reconstruction is the ability to preserve the shape of the building boundary during reconstruction. However, the density of edges recovered using existing detectors has not reached a level sufficient to recover full closed boundaries of objects. This limits the exclusive use of edge features for 3D boundary reconstruction. 3D boundary reconstruction from edge features involves three processes namely edge detection, boundary detection and reconstruction, and edge matching to recover 3D building geometry. With edge detection, fragmentation and missing edges hinder the successful recovery of complete object boundaries. Furthermore, the need for thresholding parameters for edge detectors contributes to a reduced number of detected edge pixels, resulting in piecewise open boundaries. A 3D reconstruction from recovered fragmented lines results in a sparse 3D model. Based on the challenges presented above, the research presents a novel edge detection and building boundary reconstruction framework from satellite images. By recovering dense edge features, detailed 3D wireframe models are produced. The proposed solution works on the idea that edge information extracted from different detectors with parameter variation can be aggregated to obtain better edges. The recovered edges ultimately result in improved boundary detection and reconstruction. The solution was achieved through two objectives. The first objective was to develop a

sub-framework for edge detection that reduces fragmentation and missing edges through aggregation. The second objective was to develop a boundary detection and reconstruction sub-framework from the detected edges. From the first objective, an Accumulator Based Edge Detector (ABED) framework was developed that aggregates edges by automated parameter tuning. Good edge localisation is maintained by a localisation filter during edge aggregation while keeping salient edges. ABED introduces genericity since gradient operator agnostic. From the dense edge map produced by the edge detection sub-framework, a building boundary shape reconstruction sub-framework was developed termed Accumulator Based Line Detector (ABEDL). The boundary recovery method works by recovering longer unbroken lines through a Breath-First Search (BFS) line search algorithm with parameter relaxation. The boundaries are then matched to produce detailed 3D building wireframes. To evaluate the frameworks, ten different building sites were selected from five different locations around the world. The sites were chosen because of different building architectural styles, differing imaging resolution, and varying view angles. For benchmarking, ABED was compared against Parameter Free Canny Edge Detection (CannyPF) [1] and Parameter Free Edge Drawing (EDPF) [2] for each site. ABED recovered more edge pixels with improved recall for all ten sites. Furthermore, the Root Mean Square Error (RMSE) for edge position errors was less than 1.4 pixels when compared with ground truth images. ABEDL was benchmarked against eight state-of-the-art line detectors namely Edge Drawing Lines (EDLines) [3], Line Segment Detector (LSD) [4], Canny Lines (CannyLines) [1], Progressive Probabilistic Hough Transform (PPHT) [5], Active grouping and geometry-gradient combined validation Line Detector (AG3Lines) [6], Fully Convolutional Line Parsing (F-Clip) [7], L-CNN End-to-End Wireframe Parsing (L-CNN) [8] and Hough-Transform Holistically-Attracted Wireframe Parsing (HT-HWAP) [9]. ABEDL showed good F1-Score on all ten sites. Furthermore, ABEDL recovered more boundaries for all sites. The major finding of the developed frameworks is the possibility of using satellite images to reconstruct 2D and 3D building boundaries using dense edge features as opposed to dense pixel matching. The edge detection sub-framework introduces genericity in edge detection and does not require predefined parameters. Furthermore, ABEDL presents the ability to reconstruct object form or outline and detail such as windows, allowing the reconstruction of LOD 2 and 3 detail.

Contents

List of Figures

List of Tables

Acronyms

1D One Dimension. xv, 38, 39

2.5D Two and half Dimension. xv, 25, 26, 33, 198

2D Two Dimensional. v, xv, 1, 3, 4, 6, 7, 9, 10, 12, 14, 29, 54, 59, 62, 81, 105, 106, 141, 142, 145, 190, 193, 203

3D Three Dimensional. iv, v, xv, 1–14, 16–20, 22–26, 29, 33, 37, 40, 42–44, 48, 50, 54, 55, 57–59, 61, 62, 64, 73, 80–82, 128, 129, 142, 143, 145, 189, 190, 193, 195, 199, 201–203, 206, 207

ABED Accumulator Based Edge Detector. v, xii–xv, 6, 8, 9, 76–78, 83, 85, 86, 99, 103, 105, 110, 111, 115, 145–159, 162–166, 172, 176, 179, 187–196, 198, 199, 201–204

ABEDL Accumulator Based Line Detector. v, xiii, xv, 6, 8, 9, 76, 77, 104, 105, 110, 129, 133, 134, 145, 148–150, 166, 169–172, 174–176, 178–182, 188–190, 193–195, 198, 199, 201, 202, 205, 244, 251, 256, 260

AG3Lines Active grouping and geometry-gradient combined validation Line Detector. v, xiii, xv, 68, 70, 72, 134, 168, 170–172, 178, 179, 181, 182, 193, 194

ALS Airborne Laser Scanning. xv, 2, 45

ARN Absolute Radiometric Normalization. xv, 23

B-REP Boundary Representation. x, xv, 14, 16, 30, 62, 203

BBSR Building Boundary Shape Reconstruction. xv

BFS Breath-First Search. v, xv, 9, 77, 80, 81, 131–133, 178, 195, 206

CAD Computer Aided Design. xv, 44

CannyLines Canny Lines. v, xv, 68, 72, 134, 168, 170–172, 193, 194

CannyPF Parameter Free Canny Edge Detection. v, x, xii, xiv, xv, 66, 67, 70, 71, 147–152, 154, 156, 159, 163, 164, 191

CBD Central Business District. xiii, xv, 154, 170, 180, 237, 238, 241, 243–245, 247, 251, 256, 260

CCD Charge Coupled Device. xv

CityGML City Geography Markup Language. xv, 14

CNN Convolutional Neural Network. xv

DCNN Deep Convolutional Neural Network. xv, 44

DL Deep Learning. xv, 44–46

DLT Direct Linear Transform. xv

DSM Digital Surface Model. iv, x, xv, 5, 8, 13, 25–27, 30, 33–38, 42, 43, 45, 49, 51–55, 59, 61, 73, 74, 203, 206

DTM Digital Terrain Model. xv, 49, 53

DWT Discrete Wavelet Transform. xv, 50

ED Edge Drawing. xv, 66, 71

EDLines Edge Drawing Lines. v, xv, 50, 68, 71, 72, 103, 104, 108–110, 134, 168, 170–172, 181, 193, 194

EDPF Parameter Free Edge Drawing. v, x, xii, xiv, xv, 67, 147–152, 154, 156, 159, 163, 164, 176, 177, 191

EO Exterior Orientation. xv

F-Clip Fully Convolutional Line Parsing. v, xv, 168, 169, 171–173, 193, 194

GCP Ground Control Point. xv, 23, 42

GPS Global Positioning System. xv, 21

GSD Ground Sampling Distance. xv

GT Ground Truth. xv, 44, 162, 164, 170

HRS High Resolution Satellites. xv, 1, 2, 14, 17, 19

HRSI High Resolution Satellite Image. xv

HT-HWAP Hough-Transform Holistically-Attracted Wireframe Parsing. v, xv, 169, 171–173, 175, 193, 194

id Identification Number. xv, 131

IGV Internal Grey Variance. xv, 52

IO Interior Orientation. xv

L-CNN L-CNN End-to-End Wireframe Parsing. v, xv, 168, 169, 171–173, 175, 193, 194

LBD Line Band Descriptor. xv

LEO Low Earth Orbit. xv

LiDAR Light Detection and Ranging. xv, 2, 8, 13, 14, 55, 60, 74

LOD Level Of Detail. v, x, xv, 2, 8, 14–17, 28, 29, 31, 46, 57, 61, 203

LSD Line Segment Detector. v, x, xv, 25, 68–70, 72, 103, 104, 108, 110, 134, 168, 170–172, 193, 194, 208

MGM More Global Matching. xv, 41

MLSD Mean-Standard Deviation Line Descriptor. xv

MPM Multiple Primitive Multi-Image Matching. xv, 36

MVS Multi-View Stereo. xv

nDSM Normalised Digital Surface Model. xv, 27, 35, 45, 49, 53, 59, 75

NDVI Normalised Difference Vegetation Index. xv, 48, 53, 60

OGC Open Geospatial Consortium. xv, 14, 15, 203

OSGM Object Based Multi-Image Semi-Global Matching. xv, 40

PCA Principal Component Analysis. xv, 49

PPHT Progressive Probabilistic Hough Transform. v, xv, 103, 104, 134, 168–172, 193, 194

RANSAC Random Sample Consensus. xv, 35

RFM Rational Function Model. xv, 19

RMSE Root Mean Square Error. v, xii, xv, 162–164, 169

RPC Rational Polynomial Coefficients. xv, 11, 19, 22, 23, 42, 77, 142, 143

RRN Relative Radiometric Normalization. xv, 23

SGM Semi-Global Matching. xv, 25, 34, 35, 38, 40, 41, 59

SIFT Scale Invariant Feature Transform. xv, 142

SLAM Simultaneous Localisation And Mapping. xv, 196

TIN Triangulated Irregular Network. xv

USA United States of America. xv

1 Introduction

Population and economic growth in recent decades have resulted in increased urbanisation. Consequently, there is a marked increase in urban infrastructure in different regions of the world [10, 11, 12, 13]. This rapid increase has resulted in the need to sustain urban growth, as evidenced by the emergence of the concept of smart cities [14, 15]. The improved ability to collect data and information from multiple devices at varying scales has enabled efficient planning and management of urban areas [16]. In addition, mitigation activities, such as disaster management, require detailed information on the built environment for timely decision making [17, 18, 16]. An often critical requirement is the production of up-to-date maps to enable planning and tracking of urban growth and change. More recently, up to date Three Dimensional (3D) maps of urban environments are required to complement traditional Two Dimensional (2D) maps.

Inevitably, 3D models have become a preferred means of modelling urban environments. 3D models offer various advantages over traditional 2D mapping. Some of the advantages are a better and realistic representation of the earth's surface, realistic visualisations of urban landscapes, and support for applications that use the third dimension [19, 20, 21]. Some applications areas that make use of 3D data are radio network planning, disaster managements, solar potential estimation, energy demand estimates, population estimation, navigation and gaming [16, 22, 23, 24, 25].

One of the main data sources used in large area 3D building map production is imagery acquired from airborne platforms such as satellites and aircraft. The need for frequent map updates has driven the demand for data acquisition platforms with high temporal resolution and large acquisition swaths. Space-based satellite imaging has provided a solution for high-temporal mapping, in particular High Resolution Satellites (HRS). Technological advancements have accelerated both the production and launches of a plethora of HRS satellites [26, 27, 28, 29, 30]. Earth observation satellites have proven to be a viable solution to the frequent and accurate monitoring of the evolving urban landscape. High-resolution sensors, such as WorldView, SPOT, Pleiades, and SuperView are some of the popular imaging sensors [31, 29, 30, 32]. For example, the

WorldView 3 satellite can acquire imagery at a resolution of $0.31m$ with a minimum swath width of 20km [33, 34]. Although the swath of High Resolution Satellites (HRS) had been the limiting factor to frequent large area acquisition, the emergence of constellations has increased acquisition coverage. Resultantly, the temporal resolution for image acquisition has been further reduced. Most HRS are now capable of a two-day revisit period [27, 28, 35, 29, 30]. This has allowed for the production of frequent map updates. With the increase in level of detail in imagery and acquisition frequency, the new challenges have moved from data availability to efficient and accurate data processing [36, 37, 38, 39]. In addition to the high-detailed images acquired by these sensors, agile imaging platforms allow multiview scene acquisition. Variable viewing angles make way for multi-view stereo imagery and ultimately, 3D model production [40, 41].

An alternative data source for the production of large scale 3D building models is Light Detection and Ranging (LiDAR) acquired from airborne platforms, typically termed Airborne Laser Scanning (ALS). ALS has the advantage of acquiring direct 3D information about the landscape [42, 43]. In addition, LiDAR provides intensity measurements that help with object classification and filtering tasks. ALS has the ability to acquire data over cities at very high resolution compared to satellites [44, 45]. However, ALS cannot be flown as frequently compared to satellites due to flight planning requirements and weather restrictions. Satellite imagery has proven to be a viable alternative when LiDAR is not readily available.

The modelling of 3D urban landscapes from satellite images has received considerable attention [46, 47, 48, 40, 49, 41, 50, 51]. The goal is to reconstruct 3D models with higher Level Of Detail (LOD) and accuracy [52, 53, 54, 55]. With the ever increasing technological advancements, this trend is set to continue in the future, hence the need for ongoing research in image information extraction and modelling from continuously acquired satellite imagery. With 3D building models, the objective is to improve both the fidelity of the building boundaries and the LOD recovered.

3D building models are produced using multiview images of the same area with varying off-nadir angles. Agile imaging platforms on HRS allow for the capture of stereo or multiview imagery both along and across track in short imaging intervals [31, 29].

Having multiple image views has provided solutions for various 3D model production challenges, namely occluded views, pixel matching ambiguities, and improved accuracy in 3D point determination [50, 48]. Consequently, new research topics have emerged in automated extraction of 3D information from single and multiview satellite images efficiently. Most of the work has been aimed at developing automated processing algorithms for 3D information extraction [56, 57, 46]. The processing algorithms aim to be sensor-agnostic with the ability to handle large data volumes. The increased processing power of computing resources has helped in the continued research in this area.

Complimentary to 3D building model production is to ensure the correctness of the modelled 3D buildings. While the rapid production of these models is now possible, there is a growing need to improve object shape representation of the produced models. Efficient processing requirements have not compromised the need for accurate object shape representation of detected objects. Accurate object shape extraction has been mainly focused on building boundary shape detection, since buildings are the majority feature in urban landscapes [58, 59, 60, 61]. Accurate building boundary shapes are required for servicing both 2D and 3D applications. Consequently, the area of building boundary shape extraction has also remained an area of active research.

Urban landscapes are composed of a plethora of structures with differing shapes and sizes that require accurate modelling [61]. Recovering 3D building models from satellite images has its challenges. Urban scenes are composed of complex volumetric shapes, which are difficult to model explicitly in the image domain. The problem of recovering these 3D building boundary shapes from satellite imagery has been largely solved by using radiometric analysis approaches. From stereo or multi-view images, point clouds are produced by establishing pixel correspondences between images using correlation methods [62, 57, 63, 56, 48]. However, correlation-based methods are mainly point or area processes, and thus problems arise in deriving exact building boundary shapes from the corresponding point or patches [42, 49, 64, 61]. Building boundary shapes must then be recovered from point clouds often as a post-process. However, challenges with boundary retrieval from image derived point clouds, namely matching at reduced resolution, inability to preserve sharp discontinuities at object boundaries and point mismatches, result in boundary shapes of reduced accuracy and fidelity.

With the recovery of building boundary shapes, the challenge is to develop automated pipelines that model building boundary shapes permissible by the imaging resolution. While machine learning-based methods have had a significant impact in building detection, overreliance on accurate and large training data sets presents a bottleneck in learning all available variations of building boundary shapes [46, 65, 66]. The lack of explicit representation for all geometric shapes that make up man-made structures implies most building boundaries are recovered by some level of estimation and a-priori information such as rectilinear constraints and common linear arrangements of object boundaries [65, 66, 58, 59, 60, 61]. The main challenge herein lies in having the ability to accurately recover building boundary shapes using semantic knowledge rather than explicit mathematical representation.

As mentioned above, 3D information recovery is largely a point matching process. From the points, boundary reconstruction methods require a boundary detection step after which the recovery of the boundary shape is attempted [65, 66, 67, 58, 47]. An alternative 3D reconstruction approach that has the potential to improve 3D building boundary shape reconstruction is the use of edges to recover 3D object boundaries. Edges contain more information on object structure, orientation, and size. This makes them ideal for boundary shape recovery [68]. Consequently, the key to any successful estimation of building boundary shapes is an efficient edge detection step [69, 1, 6]. The direct use of edges for the recovery of 3D building shapes has been limited due to the challenges encountered with edge detection, specifically in satellite imagery. Edge fragmentation and missing edges contribute to the difficulties of using edges for full 3D reconstruction. Weak lighting conditions, sensor dynamic range, weak contrast between structures, and image noise result in fragmented and missing edges during boundary retrieval [70, 71, 72]. Consequently, the resulting 2D and 3D boundaries are estimated from incomplete edges, resulting in sparse line models. As a result, edges have been used mainly as support features for other tasks such as pixel and feature matching [73, 74].The work presented in this research aims to alternatively reconstruct 2D and 3D building wireframes from edges using satellite imagery.

1.1 Problem Statement

Building boundary shape recovery methods aim to recover boundaries of high fidelity. One of the main sources of the production of 3D building models is satellite imagery. Point clouds produced from satellite stereo or multiview imagery are used in inferring building boundary shapes in urban scenes. However, the resulting shapes are limited in accuracy because of a lack of accurate boundary information about the object in the point data. In addition, creating a point cloud requires dense pixel matching, which does not preserve object boundary information. Boundary information is often retrieved by edge detection in the input images. The recovery of complete unfragmented edges from an image has remained a challenge [6, 75, 71]. Fragmentation of edges and missed edges limit the use of edge features to reconstruct the shape of the boundaries. Consequently, edges are used rather as support features for other processes, such as point and feature matching, to reduce matching ambiguities and preserve DSM break lines rather than for full reconstruction of the 3D building boundary shape. This is because the result of 3D edge-based reconstruction from satellite images is a sparse line model, as opposed to a detailed 3D building boundary model [74]. In an effort to solve issues arising from missing or fragmented edges, other non-edge detection methods, namely semantic segmentation and object-based image classification, are used to further enhance building boundary detection and recovery [68, 76, 77]. For the methods that attempt to use edges for reconstruction, linear edges are mostly used [3, 78, 1]. This introduces a limitation when recovering complex non-linear shapes. This is because methods that use linear edge features operate under the assumption that objects in the scene are mainly composed of rectilinear line segments [79, 80, 81]. This limits boundary recovery methods to rectilinear structures at the expense of curved structures. In addition to fragmentation, the lines are not attributed to any building or boundary. Furthermore, the detection of parameterised lines often requires a minimisation approach, which ultimately limits the number and length of lines detected [3, 6, 1, 82]. Therefore, the result is a sparse 3D line model. In addition to existing challenges in edge detection, the lack of a generic edge detector presents another challenge because different edge detectors require different parameters that vary according to image quality. The lack of global parameters for edge detectors contributes to fragmentation and missed edges. Consequently, complete reconstruction

of the 3D buildings based on edges has remained a challenge. Recently, deep learning methods have been proposed for line segment detection [7, 9] and wireframe reconstruction [8, 83, 84] from images. Although deep learning methods perform better on datasets that have geometry that resembles that found in the training dataset [84, 85, 83], their adaptability to different scene contexts remains a challenge. Most deep learning methods for line detection and wireframe reconstruction have been trained on mainly terrestrial images [83, 7]. There is a lack of publicly available training data for satellite images, which presents a challenge [86, 87]. In addition, the need to train deep learning models for different imaging resolutions and scene contexts [84], adds to the challenges of adopting deep learning methods for satellite image data. For example, Zhou et al. [84] uses 6300 and Zhou et al. [85] uses 23000 images for training and validation. This amount of data required for training adds to the limitations. Consequently, generic methods are still required for varying scene contexts.

1.2 Aims and Objectives

1.2.1 Research Aims

The aim of this research is to develop an edge-based boundary reconstruction framework that improves the resulting building boundary shapes. The goal is to preserve building form (exterior building outline) and detail (interior objects such as windows). The developed framework is made up of two sub-frameworks, namely an edge detection framework (ABED) and a building boundary shape reconstruction framework (ABEDL). ABED aims to reduce fragmentation and missing edges. ABEDL reconstructs 2D and 3D building boundary shapes using edges.

1.2.2 Objectives

The aim of the research is achieved through the following objectives:

1. Develop a novel generic edge detection framework that aggregates edges detected using multiple detectors with automated parameter tuning. This development has the following sub-objectives:

- Investigate the use of multiple edge detectors for edge aggregation while maintaining good localisation
- Investigate the performance of the developed framework under varying imaging conditions without the need for parameter tuning.
- Establish the framework's ability to recover a dense edge network of unbroken edge chains representing object form (external boundary) and detail (interior facade features).

2. Develop a novel building boundary shape reconstruction framework that retrieves long unfragmented curves from the edges detected by the edge detection framework. This is achieved through the following sub-objectives:
   - Develop a line and curve detection algorithm from a dense edge map.
   - Develop a boundary shape reconstruction algorithm that uses lines as input and creates longer unfragmented building boundaries.
   - Show the feasibility of using the reconstructed lines and curves in 3D reconstruction.

1.2.3 Research Questions

The research questions to be answered in addressing objective 1 and 2 are:

1. Using multiple edge detectors presents a challenge because the edge position recovered by each detector is different. Merging edges from different detectors often results in double edge effects and dangling edges. In developing the edge detection sub-framework, how can multiple edge detector results be fused or aggregated to produce long unbroken edge chains that converge to the same position (localisation) using automated parameter tuning?
2. Edges are often fragmented because of weak contrast, shadows, and image noise. Does edge aggregation with automated parameter tuning improve edge detection in dark regions, in the presence of shadows and low contrast regions?
3. By reducing fragmentation and missing edges, does this result in improved boundary shape recovery using edges in both 2D and 3D, and ultimately results in improved

3D building wireframe models using the developed boundary shape reconstruction framework?

4. Pixel based reconstruction from satellite images often achieves LOD 1 buildings. Can higher LOD be achieved from satellite images using edges?

1.2.4 Hypothesis

1. Edge aggregation results in improved edges with reduced fragmentation.
2. Boundary reconstruction can be achieved through graph traversal using dense edges with reduced fragmentation.

1.2.5 Methodology

Challenges in retrieving building boundary shapes often result in reduced model accuracy. Most of these challenges arise from the failure to fully recover object form and detail when creating 3D models. Most algorithms aim to recover boundaries from derived products such as DSMs or LiDAR point clouds[88, 89, 90]. For very high-resolution point clouds, this has proved successful [25, 42]. However, this does not apply to DSMs from imagery since edge information is not accurately preserved during image correlation. The research proposed here aims to improve object boundary detection, which will ultimately result in improved 3D building boundary shapes. The proposed workflow followed is depicted in Figure 1.

The methodology begins with a review of building reconstruction strategies in which the current challenges and gaps are identified. From the identified gaps, the problem statement is formulated. Based on the identified problem, data preparation, where relevant image scenes are selected for various sites to use in the research follows. After data preparation, the next step is the development of the edge detection sub-framework termed ABED. The edge detection framework detects edges with reduced fragmentation and missing edges. The output of the edge detection framework is a dense edge map.

After edge detection, the next step is the development of the building boundary shape reconstruction sub-framework termed ABEDL. This framework is composed of two parts, namely boundary detection and reconstruction, and building boundary shape

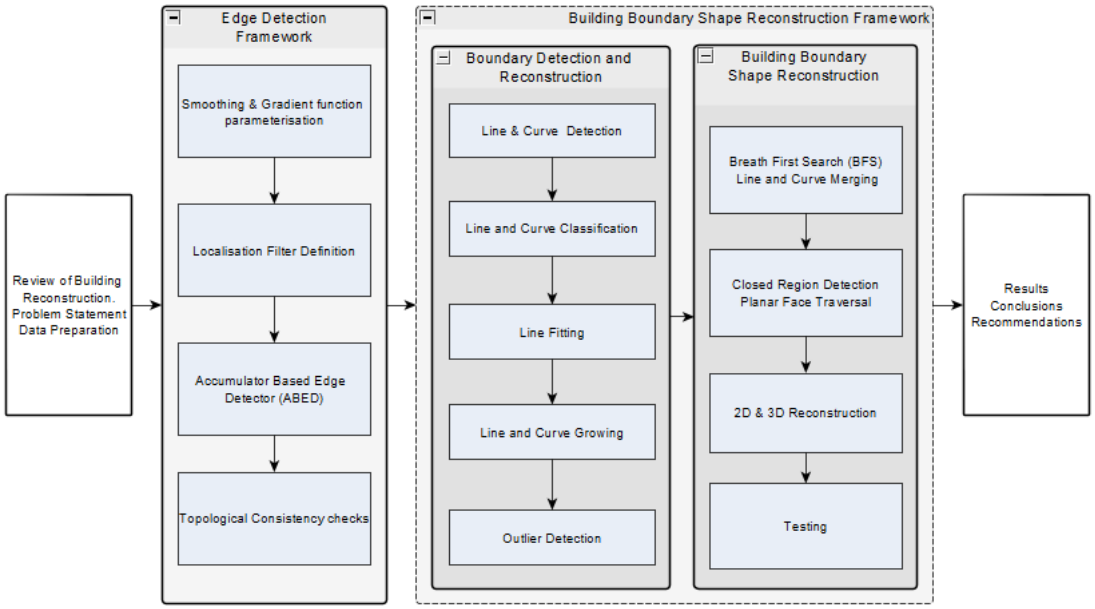


Figure 1: Proposed Research Methodology workflow. The methodology begins with a review of building reconstruction techniques and problem definition followed by the method, comprised of two frameworks and then ends with results and conclusions.

reconstruction. Using the dense edge map as input, lines and curves that constitute building boundaries are detected. Finally, the recovered lines and curves are used to estimate the boundary shapes of buildings in the images. Connected boundaries are recovered using a Breath-First Search (BFS) algorithm with parameter relaxation to reduce fragmentation and improve line orientation. The lines are used to recover closed boundaries using a planar face traversal algorithm in 2D. In the case of 3D building boundary shapes, multiview scene boundaries are used to recover 3D building wireframes.

The results of each of the frameworks are then evaluated qualitatively and quantitatively. First, the results from the ABED are presented, followed by the results of the ABEDL. From the results, a discussion of the results is given followed by conclusions and recommendations.

1.2.6 Contributions

The outcomes of this research are:

- A novel and generic edge detection framework for satellite imagery that retrieves longer object edges. The edges recovered improve the building boundary shapes

in both 2D and 3D. The edge detection framework has various advantages that are desirable for any edge detection task. The framework implements automated parameter tuning for smoothing and gradient thresholds. Consequently, the task of providing optimal smoothing and threshold parameters is eliminated by using an accumulator that operates in a wide parameter space. In addition, an edge detection framework that is operator-agnostic and adaptable to any edge detection task is developed. A major contribution is the proof that dense edge features, capturing both object form and detail, can be recovered with good localisation and ultimately enhance object shape retrieval. The novelty herein lies in the ability to preserve both object form and detail during edge detection.

A novel line and curve retrieval framework that improves line orientation while detecting more linear segments, including short line segments. Current line detection methods use a minimum line length detection parameter, which filters short edges, resulting in fragmentation and more missed edges. However, the framework presented in this research recovers connected lines and curves from dense edge features with reduced fragmentation.

- A line recovery algorithm that uses global line segment information to improve line orientation as opposed to local information used with current state-of-the-art line detectors.

- An edge-to-building shape reconstruction strategy that reconstructs building form and detail using line segments. Curve and edge features are ubiquitous in any computer vision task. Most algorithms focus on straight line retrieval. The pipeline proposed here improves line detection from images, while adding the ability to detect connected lines that form curves.

- A 3D reconstruction pipeline that uses edges instead of pixels to reconstruct 3D building wireframes. With enough edge density, the building boundary shape is retained in the 3D output. An edge-based wireframe reconstruction procedure provides an alternative to exhaustive pixel matching, while preserving shape information.

1.3 Scope of Research

The research is limited to the reconstruction of buildings from satellite images with a resolution ranging from $0.5m$ up to $0.3m$. Multiview satellite images of five urban cities are used in the research. Object boundary shape detection and reconstruction are limited to building structures. Line and edge matching are not part of the research.

1.4 Assumptions and Limitations

Assumptions made in this research are:

1. The sensor model employed is the Rational Polynomial Coefficients (RPC) camera model.
2. All sensor models used in the satellite images have been adjusted and corrected for biases introduced during imaging conditions and are ready for use.
3. The research is limited to high resolution satellite imagery of resolution $0.5m$ up to $0.3m$. The reconstruction of the boundary shapes is limited to building structures.
4. Multiview images are used in the 3D boundary reconstruction framework.
5. Existing line matching algorithms are used to reconstruct 3D models.
6. Recovery is used to refer to reconstruction

1.5 List of Publications

- Mapurisa W. and Sithole G., "**Improved edge detection for satellite images**," ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 2, pp. 185–192, 2022.

1.6 Thesis Structure

Chapter 1 gives an introduction and motivation for this research. This chapter details the purpose and objectives of the study together with the limitations and scope.

Chapter 2 outlines concepts of 3D reconstruction and building boundary shape recovery from images. The current steps followed in 3D reconstruction are detailed. In addition, the chapter introduces some core concepts that are used throughout the research.

Chapter 2.5 gives a detailed review of current reconstruction strategies. From this review, challenges and gaps in building boundary shape reconstruction are identified. From the identified gaps, the methods used in this research are formulated.

Chapter 3 gives a brief description of the method developed in this research. A brief description of the data used and a summary of each of the steps followed in the research are given.

Chapter 4 presents the edge detection framework developed to retrieve building edges. Details of the implementation that result in improved edge detection are given.

Chapter 5 details the first part of the building boundary shape reconstruction framework, which is the detection and reconstruction of boundaries as a set of lines and curves. The boundary detection and reconstruction strategy uses edges as input.

Chapter 6 implements the second part of the building boundary shape reconstruction framework, which is building shape reconstruction where the lines and curves detected are connected to form the resulting shapes of buildings in 2D and detailed 3D wireframes.

Chapter 7 presents qualitative and quantitative results of the edge detection framework and the building boundary shape reconstruction framework.

8 provides a discussion of the results presented in Chapter 7.

Chapter 9 presents the conclusions drawn and recommendations.

2 Background and Literature Review

2.1 Introduction

This chapter discusses concepts of 3D reconstruction from multiview satellite images. Before discussing the literature review in Section 2.5, an appreciation of the building reconstruction pipeline is required. Although there exists other data sources for 3D reconstruction such as aerial imagery and LiDAR, the discussion in this chapter is limited to reconstruction using satellite image data. Firstly, an introduction to 3D city modelling is given. This is followed by a description of the workflow typically followed for creating 3D models. The workflow starts with data acquisition, which is described in Section 2.2.1, followed by a discussion of the image pre-processing steps required with satellite images in Section 2.2.2. Feature extraction, image matching, and 3D information extraction are discussed in Sections 2.2.2.4, 2.2.2.5 and 2.5.1.2 respectively. DSM generation, object detection, and building reconstruction are discussed in Sections 2.2.3, 2.2.4 and 2.2.5. Finally, Sections 2.3 and 2.4 introduce some of the concepts used in this research.

The chapter goes on to further give a detailed review of Building detection and reconstruction methods from high resolution satellite imagery. Part of the requirements in undertaking this research is reviewing current methods with regard to building shape recovery, with the aim of identifying current challenges and gaps. From the gaps, new areas of improvement are identified and a new method is formulated. Section 2.5 reviews the main aspects related to object detection and building reconstruction from high-resolution satellite images. This review will set the scene for the method presented in Chapter 3. With the exception of LiDAR, building object detection and reconstruction use two main inputs, namely satellite images or DSMs or a combination of both. Methods for DSMs are discussed in Section 2.5.1. A discussion on building detection from these sources is given in Section 2.5.2, followed by the subsequent reconstruction in Section 2.5.3. A summary of the algorithms reviewed is given in Section 2.6. Section 2.7 gives a review of edge and line detection and the challenges that exist, since they are frequently used in building reconstruction. From these reviews, the challenges and gaps are identified and summarised in Section 2.8.

2.2 3D City Modelling

3D city modelling is the process of creating a digital representation of the man-made and natural features of urban or city areas [16, 88]. These features may include buildings, roads, trees, street furniture, and bridges [18, 91]. 3D models are created from various sources, namely satellite stereo images [92, 89], aerial imagery [93] and LiDAR [88] data. Since the research presented here focuses on the creation of 3D models from satellite images, the discussion is limited to the workflow of creating models from HRS images. HRS imaging platforms' agility, allows for off-nadir imaging of the same area, resulting in multiview scenes[28, 27, 32]. Multiple image views solve pertinent challenges in reconstruction, such occlusion, shadows, geometric distortion, and matching ambiguities [50, 57]. This has resulted in improved 3D information recovery from satellite images. Figure 2 shows the typical workflow that is followed when creating 3D models from multiview satellite imagery. Stereo or multiview images allow for 3D measurements up to the pixel level [57], and this results in more detail recovered. Current trends predict the launch of more HRS with ever improving resolution.

Three-dimensional models are created at various Level Of Detail (LOD) for rendering. LOD is used to differentiate the representation of the model at different scales and detail. LOD, according to the CityGML Open Geospatial Consortium Specifications (OGC), range from LOD 0-3 [91]. LOD 0 represents points, 2D building footprints or roofprints. LOD 1 represents building footprints or roof outlines extruded to the ground, alternatively called block models. LOD 2 are building models with standard roof structures such as domes, ridges, or peaks. Finally, LOD 3 are building models with both roof structures and facade features such as windows, doors. In other words, these are detailed 3D meshes of the exterior of the building [94]. Figure 3 gives a representation of different LODs. Column one shows the OGC LOD specification and the other columns are refined definitions by Biljecki et al. [94] to accommodate more details of the buildings.

The produced 3D models are often presented as Boundary Representation (B-REP) models or 3D meshes that represent the outer surface boundary of objects. B-REP models are made up of vertices, lines, and facades that are connected to represent the

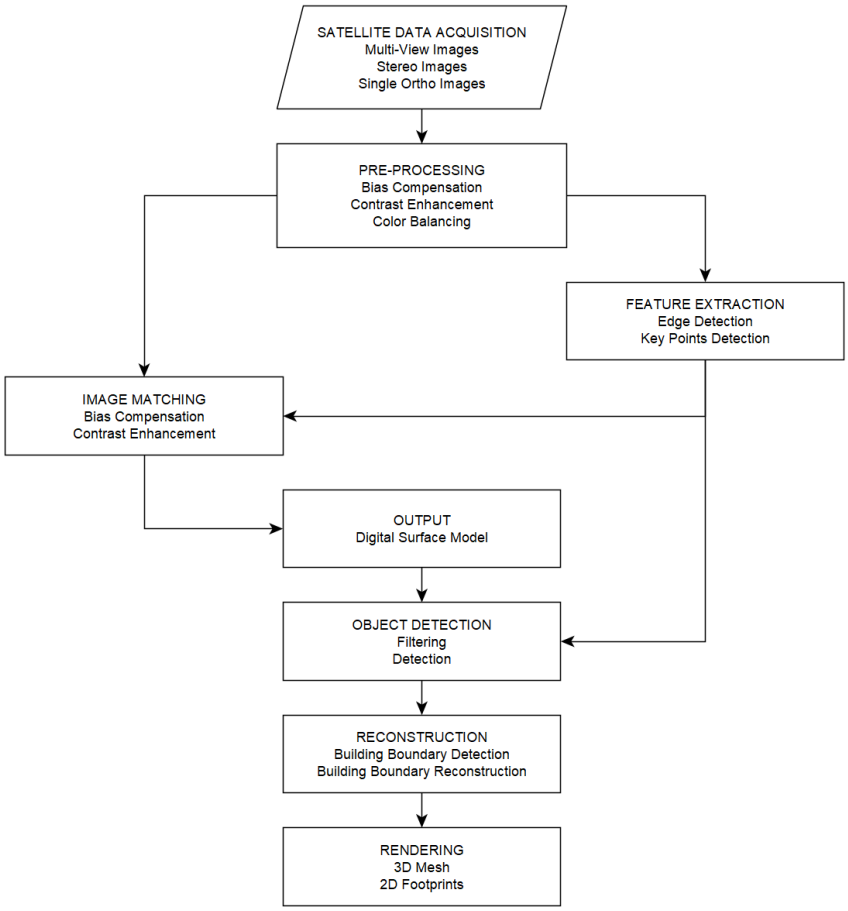


Figure 2: 3D Building reconstruction workflow from Multiview Satellite Images.

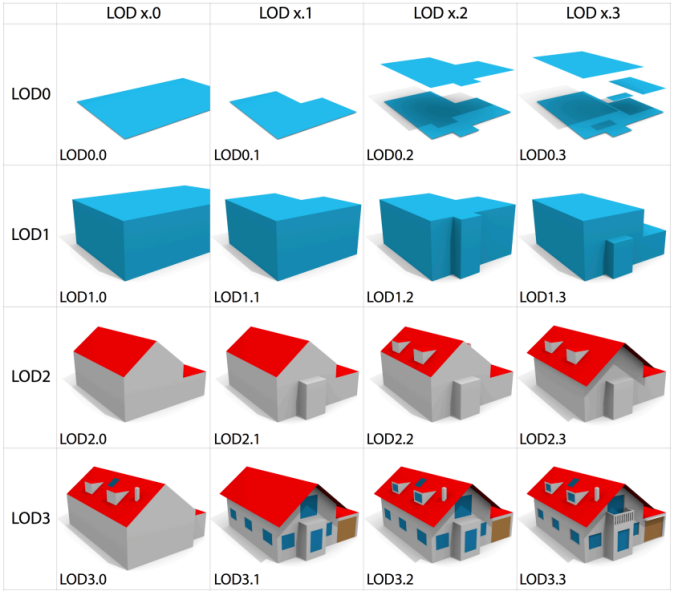


Figure 3: Refined LOD representation of building structures presented by [94]. Column one represents the OGC specified LODs while the other columns represents the refined LODs by introducing sub LOD categories [94].

outer building surface shape [95]. The models shown in Figure 3 are B-REP models of buildings. Example of an LOD 2, B-REP model of a house is illustrated in Figure 4. For this research, the goal is to recover connected lines and closed boundaries that constitute LOD 1 and 2. These boundaries are then used to recover 3D information. This is different from the traditional approach of estimating the footprint of the building from a point cloud and extruding the footprint using a height to obtain the desired model [46, 92, 66]. Figure 5 shows examples of some LOD 2 boundaries that the proposed framework aims to recover.

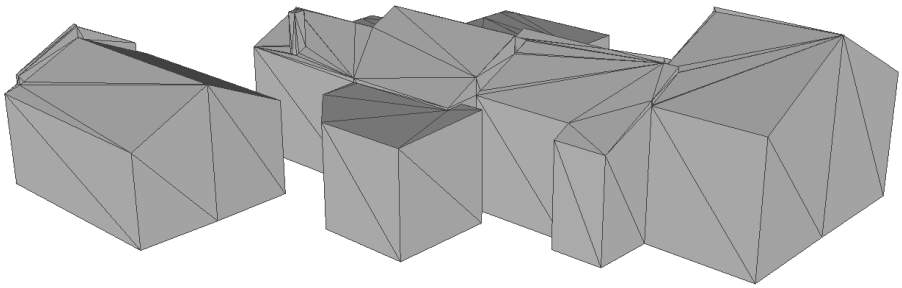


Figure 4: Boundary Representation (B-REP) Model of a house rendered using vertices and triangles. The outer building surface shape is depicted.

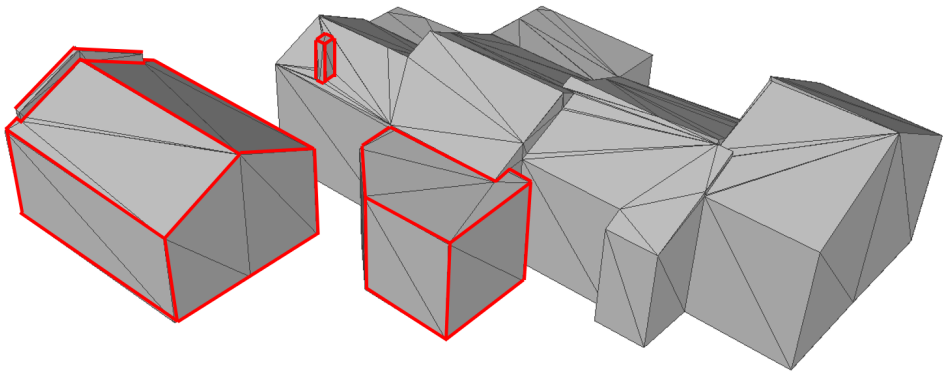


Figure 5: Boundary Representation (B-REP) Model of a house. Example 3D boundaries in red that the proposed framework aims to recover.

The extraction of 3D information from stereo or multiview images has become one of the most cost-effective methods for large area 3D mapping. With the ever increasing resolution, the capabilities of existing algorithms are constantly being improved [50, 66, 96]. Improved feature extraction, feature matching, object identification, and

classification are constantly being explored [97, 98]. As imaging resolution increases and approaches ground-based camera resolutions, the applicability of computer vision algorithms to space-borne sensors is constantly being investigated [97, 99, 55]. This will assist in leveraging the robustness of existing computer vision algorithms. The research presented here aims to achieve improved object shape retrieval from high-resolution satellite images. With satellite data, research is required to improve the shapes extracted for building boundaries. The requirements for 3D models have moved from extracting LOD 1 objects to accurate shape and object detail extraction. The motivation for this research is to improve the extraction of building boundary shapes and facade details (higher LOD) from satellite images [66, 92, 60, 100, 58]. The following sections discuss the various stages for creating 3D models from HRS as depicted in Figure 2.

2.2.1 Satellite Data Acquisition

Satellite data acquisition generally begins with a tasking request, where the area to be imaged is identified and the purpose of the imagery is defined. Depending on the application, the satellite acquires image data covering the area of interest within hours or a few days of the request if the weather conditions are favourable. Since HRS have a short revisit time of less than four days [30, 29, 31], weather does not present a huge challenge as the satellite uses frequent weather updates to acquire scenes. For 3D reconstruction, images with the appropriate base-to-height ratio are required for 3D information extraction. The appropriate base-to-height ratio is achieved by off-nadir imaging of the area of interest [101, 102]. The off-nadir angles that give the appropriate base-to-height ratio for 3D reconstruction are between 15^o and 30^o [32, 31, 28]. The agility of the sensor platform and constellations have ensured a system that works in tandem to achieve rapid stereo acquisition. The imaging interval of the multiview scenes is limited to seconds, which reduces radiometric variability in the acquired scenes. Large radiometric variability presents some challenges during feature matching. A minimum of two scenes are captured for 3D mapping purposes. The improved radiometric resolution of HRS [34] has ensured the capture of more features in images, which benefits the reconstruction of 3D objects. The single and multi-view scenes captured by HRS are the primary input for building detection and 3D reconstruction.

These scenes are used for manual 3D capture, automated image matching, and automated building detection methods after some pre-processing. The figure below shows an example satellite scene of Calgary, Canada, at a $0.5m$ resolution.

Figure 6: Sample of a satellite image acquired over Calgary, Canada at resolution 0.5m. Imagery courtesy of Luxcarta [103].

2.2.2 Image Pre-Processing

Raw satellite imagery undergoes pre-processing to account for inherent errors that arise from the imaging process [104, 105, 106, 107]. Pre-processing removes unwanted artefacts and enhances the images for downstream processing. The main types of errors are geometrical and radiometric errors [108, 109, 110]. These errors arise as a result of varying acquisition angles, sensor attitude determination, absorption and scattering of the atmosphere, seasonal effects, and solar surface sensor interaction [108, 105, 109]. Geometrically, the images require corrections to improve ground-point determination accuracy. This process is termed bias compensation, since it removes unwanted positional biases introduced during imaging [104, 108, 111]. With regard to radiometry, radiometric normalisation is required, especially in the case of multiview images. Differences in image acquisition times result in radiometric disparities in the images

[109, 110]. This often affects the image or feature matching process which relies on radiometric similarity measures. Radiometric normalisation thus removes errors arising from atmospheric conditions or sensor radiometry, ensuring radiometric consistency across scenes. Radiometric normalisation ensures improved matching accuracy in multiview images when recovering 3D information. Bias compensation and radiometric normalisation is discussed next, since it has a direct impact on 3D point and feature determination. Since bias compensation is critical in the accuracy of determined ground positions, the various models used in correction satellite images are also discussed for completeness in Section 2.2.2.2. The application of the models is discussed further in Section 2.5.1.2. First, a discussion of the sensor model used for HRS images is given, followed by a discussion on how errors are modelled and compensated for.

2.2.2.1 The Rational Function Model The most common sensor models for satellites are the Parallel Projection Model [112], Physical Sensor Model [113] and the Rational Function Model (RFM) [114]. The RFM relates pixel coordinates to object coordinates or vice versa [114]. The main motivation for the adoption of the RFM is the ability to model the image-to-ground relationship without camera model information or sensitive satellite metadata while achieving accuracies comparable to physical sensor models [115, 116, 104]. Consequently, satellite image vendors are not required to publish sensor-specific information to end users. Secondly, the RFM suits narrow field of view sensors found in HRS while achieving accuracies comparable with the physical sensor models. Furthermore, the performance of RFMs has been researched on extensively and their numerical stability established [104, 52, 117, 118]. An example of the RFM is the Rational Polynomial Coefficients (RPC) sensor model adopted by satellites such as WorldView [27], Ikonos, SPOT 6 and 7 [29], ZiYuan-3 [119]. The RPC sensor model is explained in this section since worldview images are used in the research. The RPC model describes the relationship between image space and ground coordinates as a ratio of two cubic functions resulting in 80 coefficients for each model [104]. The RPC model can be described by the following equations:

$$
\begin{aligned}
l_N &= \frac{P_1(\phi_N, \lambda_N, H_N)}{P_2(\phi_N, \lambda_N, H_N)} \\
s_N &= \frac{P_3(\phi_N, \lambda_N, H_N)}{P_4(\phi_N, \lambda_N, H_N)}
\end{aligned}
\tag{2.2.1}
$$

where l_N and s_N are the normalised line and sample image coordinates. ϕ_N, λ_N, H_N are the normalised latitude, longitude and height coordinates. The normalised coordinates are derived from equation 2.2.2. Normalisation scales the input coordinates to a range between -1 and 1, to improve numerical stability during 3D point determination [104, 52].

$$
\begin{aligned}
l_N &= \frac{l - lineOffset}{lineScale} \\
s_N &= \frac{s - sampleOffset}{sampleScale} \\
\phi_N &= \frac{\phi - latOffset}{latScale} \\
\lambda_N &= \frac{\lambda - longOffset}{longScale} \\
H_N &= \frac{H - heightOffset}{heightScale}
\end{aligned}
\tag{2.2.2}
$$

P_n is the cubic polynomial of the form:

$$
\begin{aligned}
P_1 = F(\phi_N, \lambda_N, H_N) = a_1 + a_2.\lambda_N + a_3.\phi_N + a_4.H_N + a_5.\lambda_N.\phi_N \\
+a_6.\lambda_N.H_N + a_7.\phi_N.H_N + a_8.{\lambda_N}^2 + a_9.{\phi_N}^2 + a_{10}.{H_N}^2 + a_{11}.\lambda_N.\phi_N.H_N \\
+a_{12}.{\lambda_N}^3 + a_{13}.\lambda_N.{\phi_N}^2 + a_{14}.\lambda_N.{H_N}^2 + a_{15}.{\lambda_N}^2.\phi_N \\
+a_{16}.{\phi_N}^3 + a_{17}.\phi_N.{H_N}^2 + a_{18}.{\lambda_N}^2.H_N + a_{19}.{\phi_N}^2.H_N + a_{20}.{H_N}^3
\end{aligned}
\tag{2.2.3}
$$

$$
\begin{aligned}
P_2 = F(\phi_N, \lambda_N, H_N) = 1 + b_2.\lambda_N + b_3.\phi_N + b_4.H_N + b_5.\lambda_N.\phi_N \\
+b_6.\lambda_N.H_N + b_7.\phi_N.H_N + b_8.{\lambda_N}^2 + b_9.{\phi_N}^2 + b_10.{H_N}^2 + b_11.\lambda_N.\phi_N.H_N \\
+b_12.{\lambda_N}^3 + b_13.\lambda_N.{\phi_N}^2 + b_14.\lambda_N.{H_N}^2 + b_15.{\lambda_N}^2.\phi_N \\
+b_16.{\phi_N}^3 + b_17.\phi_N.{H_N}^2 + b_18.{\lambda_N}^2.H_N + b_19.{\phi_N}^2.H_N + b_20.{H_N}^3
\end{aligned}
\tag{2.2.4}
$$

$$
\begin{aligned}
P_3 = F(\phi_N, \lambda_N, H_N) = c_1 + c_2.\lambda_N + c_3.\phi_N + c_4.H_N + c_5.\lambda_N.\phi_N \\
+c_6.\lambda_N.H_N + c_7.\phi_N.H_N + c_8.{\lambda_N}^2 + c_9.{\phi_N}^2 + c_10.{H_N}^2 + c_11.\lambda_N.\phi_N.H_N \\
+c_12.{\lambda_N}^3 + c_13.\lambda_N.{\phi_N}^2 + c_14.\lambda_N.{H_N}^2 + c_15.{\lambda_N}^2.\phi_N \\
+c_16.{\phi_N}^3 + c_17.\phi_N.{H_N}^2 + c_18.{\lambda_N}^2.H_N + c_19.{\phi_N}^2.H_N + c_20.{H_N}^3
\end{aligned}
\tag{2.2.5}
$$

$$
\begin{aligned}
P_4 = F(\phi_N, \lambda_N, H_N) = 1 + d_2.\lambda_N + d_3.\phi_N + d_4.H_N + d_5.\lambda_N.\phi_N \\
+d_6.\lambda_N.H_N + d_7.\phi_N.H_N + d_8.{\lambda_N}^2 + d_9.{\phi_N}^2 + d_10.{H_N}^2 + d_11.\lambda_N.\phi_N.H_N \\
+d_12.{\lambda_N}^3 + d_13.\lambda_N.{\phi_N}^2 + d_14.\lambda_N.{H_N}^2 + d_15.{\lambda_N}^2.\phi_N \\
+d_16.{\phi_N}^3 + d_17.\phi_N.{H_N}^2 + d_18.{\lambda_N}^2.H_N + d_19.{\phi_N}^2.H_N + d_20.{H_N}^3
\end{aligned}
\tag{2.2.6}
$$

The normalisation parameters namely *latscale*, *lonscale*, *heightscale*, *samplescale*, *linescale*, *lineoffset*, *sampleoffset*, *heightoffset*, *lonoffset* and *latoffset* are determined by the extents of the scene as described by Tao and Hu [115] and Fraser et al. [111].

2.2.2.2 Bias Compensation in Satellite images While there are various errors associated with space imaging, there are common errors that have significant effects on the accuracy of derived ground coordinates. First of these errors are sensor attitude errors that are encountered during imaging. Sensor attitude errors are encountered in determining sensor orientation during imaging [108, 111, 104, 116]. Errors encountered in attitude determination should be accounted for during sensor modelling, as they have a direct impact on ground positioning accuracy. Sensor attitude is determined by three angles, namely yaw (rotation about the radius from the centre of the earth), pitch (rotation about the cross-track) and roll (rotation about the in-track direction) [104, 120, 108]. Grodecki and Dial [104] showed that the most significant of the three errors are pitch and roll where an error of one arc second results in a ground displacement of approximately 3.3 metres.

Ephemeris errors are the other significant sources of errors that arise from sensor position determination. On-board GPS receivers and star trackers are used on spacecrafts to determine sensor position, but they have inherent errors associated with them. The

effects of positional errors in determining the satellite ephemeris affect the accuracy of derived 3D ground points. The types of ephemeris errors that need to be accounted for can be classified into two. Firstly, in-track and cross-track ephemeris errors, which are small translations or shifts from the recorded imaging position. These small displacements affect the resulting accuracy of the determined ground point. Secondly, radial ephemeris errors, which affect the image scale, have been proven to be insignificant [104, 120]. Furthermore, drift errors have been found to be negligible for small scenes less than $50km$ long [52]. Attitude and position errors give rise to in-track and cross-track errors. These errors can be modelled as shifts or translations in image space coordinates. However, for long strips larger than $50km$, the effects of drift/time dependent errors cannot be ignored, thus a translation will not be sufficient to model the drift errors, which can cause an affine distortion [52]. For smaller imaging strips of less than $50km$, the bias compensation adjustment model is formulated as follows:

$$\begin{aligned} \delta_l &= a_0 + a_1.sample + a_2.line \\ \delta_s &= b_0 + b_1.sample + b_2.line \end{aligned} \tag{2.2.7}$$

where δ_l and δ_s, are the line and sample errors. a_0, b_0, a_1, b_1, a_2, b_2 are the error terms. As reported by Grodecki and Dial [104] and Fraser et al. [111], drift or time dependent errors remain at sub-pixel level for image strip less than $50km$. Consequently, shift terms, a_0 and b_0 in equation 2.2.7 are sufficient to model these errors. In the case where drift errors must be accounted for (i.e. strips $\geq 50km$) the above equations are modified as follows:

$$\begin{aligned} \delta_l &= a_0 + a_1.sample + a_2.line + a_3.sample^2 + a_4.line.sample + a_5.line^2 + ... \\ \delta_s &= b_0 + b_1.sample + b_2.line + b_3.sample^2 + b_4.line.sample + b_5.line^2 + ... \end{aligned} \tag{2.2.8}$$

In summary, three models can be used for bias compensation of RPCs depending on the swath of the image. a_0 and b_0 are used in the case of modelling shift errors for images with a length and swath of less than $50km$. For longer strips, a_0, b_0, a_1, b_1 model shift and time-dependent drift errors. a_1, b_1 absorb all gyro drifting errors. a_0, b_0, a_1, b_1, a_2,

b_2 allow the absorption of radial ephemeris uncertainties. Depending on image strip size, modelling up to a_n, b_n allows for all errors to be accounted for. However, equation 2.2.7 has been proven sufficient to model the most significant errors for images with a length of up to $50km$. For strips less than 100km, a_2, b_2 become significant while the rest are negligible [52, 104].

From equation 2.2.7, a_0 and b_0 can easily be solved by a single Ground Control Point (GCP) for a single image and is sufficient for small scenes. The discrepancy between the measured control points and the RPC calculated image point will constitute a_0 and b_0. Solving for a_0, b_0, a_1 and b_1 will require at least two GCPs while a_0, b_0, a_1, b_1,a_2 and b_2 will require at least three GCPs. Recent studies have shown that one GCP is usually sufficient if drift errors are insignificant and are not taken into account. The six parameters can be solved using a least-squares approach or in a bundle block adjustment [108]. In the case where these six parameters are not sufficient to solve for all positioning errors and when image strips are long RPC re-computation using GCPs is recommended [108, 104, 111]. Tong et al. [116] presents a study on the impact of using the various RPC correction models, from using only the shift terms up to the second order polynomial correction models and regenerating RPCs using computed correction terms. Equation 2.5.1.2 is used to correct RPCs using the correction models discussed.

2.2.2.3 Radiometric Normalisation The recorded radiometry by any imaging sensor is affected by atmospheric noise, viewing angle, and relief. These effects introduce errors in the measured radiometry and differences in the radiometry between different scenes. Thus, radiometric normalisation is required to account for these errors to ensure radiometric accuracy in an image and in different images of the same area [109]. Gan et al. [121] classifies radiometric normalisation into two categories, namely Absolute Radiometric Normalization (ARN) and Relative Radiometric Normalization (RRN). ARN is where the scene is corrected using measured parameters that existed during the imaging process, such as atmospheric conditions, imaging angles, sun angles, and relief information. Relative Radiometric Normalization (RRN) is where an existing reference image is used as a baseline for correcting the acquired image. The imaged radiometry is adjusted to match the reference image as closely as possible. Automated 3D information recovery is heavily dependent on the comparison of radiometry from

different images. Radiometric normalisation has an impact on image matching results. This means that large radiometric differences in the digital numbers of similar objects in the same scene result in the failure of some image matching or feature matching techniques. Consequently, these errors must be corrected before 3D reconstruction. More details on radiometric normalisation can be found in [110, 109, 121].

2.2.2.4 Feature Extraction Features in images are areas of distinct patterns or structures [122]. Features in images can be represented by points, lines or small area patches. Descriptors that describe their distinct nature compared to other features are attached to each feature. An example feature descriptor is Histogram of Oriented Gradients [123]. Example features are corners [124], edges [125], lines [74] or small image regions. Put differently, features are local descriptions of an image point or area that is different from its surrounding regions. Features, also referred to as keypoints, are often used to identify similar points or regions in different images [126, 74]. Feature detection is a processing step that is required before most image correspondence tasks. Various algorithms exist to detect and match features in satellite images [48, 53, 127]. By matching features with similar descriptors in different scenes, 3D information is recovered. Robust features are generally invariant to scale and rotation, allowing for matching in multiview scenes. Various features have been presented for image matching [128, 122, 129].

For building boundary shape detection and reconstruction, points, lines and edges are frequently used features [130, 126]. Edge detection aims to identify areas of sharp contrast changes that are usually associated with object edges. Edge features are generally used to preserve breakline information during image matching [48, 73] or to reconstruct object boundaries, such as roads and buildings [131, 67, 66]. Edge and line features provide contextual information about object presence, orientation, and size of the object. This makes edge features vital in any reconstruction process. A detailed discussion on edge and line detection is given in section 2.7.1. Figure 7 below shows results from a detector that extracts line features from a high resolution satellite image of a city.

2.2.2.5 Image Matching Image matching currently forms the core of most 3D building recovery algorithms from satellite imagery. Image matching is the process of

Figure 7: Results of Line Segment Detector (LSD) [4] on a satellite image. This is an example of line features extracted from a satellite image. The lines are characterised by fragmentation and some lines are missing.

finding corresponding pixels in multi-view images to use for 3D reconstruction [50, 56, 57]. The result of image matching is a either a Two and half Dimension (2.5D) point cloud or Digital Surface Model (DSM). A DSM represents the heights of man-made and natural features on the earth in raster format. The fidelity of the resultant DSM dictates the results of the detection of objects and shape recovery. Consequently, robust image matching algorithms are required to generate DSM or height models that are used for 3D reconstruction. The widely used algorithm for matching pixels in satellite images is the Semi-Global Matching SGM algorithm [132, 57].

2.2.3 Digital Surface Models

The most common representation of the derived 3D information from satellite and aerial imagery is a 2.5D point cloud or a DSM. DSMs represent the above sea level height of all natural and man-made objects on the surface of the earth. DSMs at city level are derived from dense image matching algorithms [57, 50, 73]. The results from the image matching process are planimetric positions of points attributed with heights. The process of creating a 2.5D point cloud or DSM from satellite or aerial imagery is illustrated in Figure 8. Given a set of multiview images, the first step in DSM production is optionally image rectification, where the images are converted to epipolar geometry [133, 134, 135, 136]. This process reduces the matching task to a one-dimensional problem by removing the y-parallax. This reduces the overall time required during the matching process [137, 138, 57, 62, 139]. Some algorithms match images without rectification, but instead employ projection-based matching techniques [48, 73, 140, 127]. The advantage of projection-based techniques is the ability to use more than two images during matching. The result of the matching process is a disparity map showing the horizontal distances between matched pixels in a given image pair. From the disparity map, 3D height information is then computed for each matched pixel pairs resulting in a DSM [49, 24]. Other than heights, DSMs generally do not contain any other information regarding the scene. Thus, different algorithms are required to extract auxiliary information from DSMs.

The resulting DSM lacks object information associated with the scene. Any lower level information such as object location, object boundaries, and shape must be derived from the DSMs. There are various algorithms that address the problem of object detection, boundary extraction, and final 3D shape reconstruction. Section 2.2.5 gives a detailed review of shape extraction algorithms. This research aims to recover detailed building boundaries as opposed to using DSMs as input.

2.2.4 Building Object Detection

Building object detection is the process of identifying building structures from images directly or from a derived DSM. The results of a building detection process are the location and extents of buildings in an image. For building detection from images and

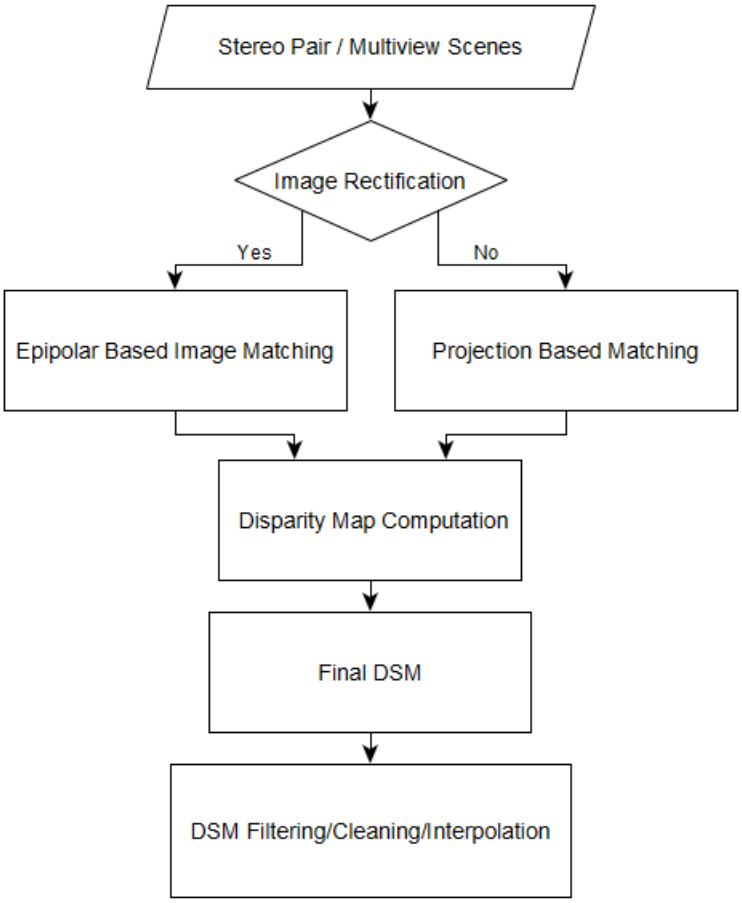


Figure 8: Digital Surface Model creation workflow from stereo or multiview satellite images.

DSMs, various approaches have been proposed [141, 142, 67, 59, 143, 144, 145]. However, the detected building object extents are not the true object boundaries, but an estimate of where the object boundary exists. Figure 9 shows the results of a building detection algorithm presented by [146]. Red regions indicate detected building structures. Detection of buildings from DSMs often requires ground filtering. Filtering removes all ground pixels or points from a DSM. This simplifies the task of identifying buildings in the resulting Normalised Digital Surface Model (nDSM) [147, 64]. Once buildings are detected, the boundary outlines are estimated from the points on the building, combining height and edge information, the extents of the building, or a combination of different image features[64, 147, 142, 148].

The goal with building object detection is to detect the correct extents of the building structure and reduce partial detections. Partial detections often result in partial and incorrect building reconstruction. Recently, deep learning methods have been extensively

Figure 9: Building detection results on a high resolution image. The red areas indicate the extents of the buildings detected by Attarzadeh and Momeni [146].

used in building object detection [46, 65, 97, 149, 150]. Semantic segmentation results of buildings from deep learning have recently been used to map large areas [151, 152]. The main limitation with building object detection is that object detail is often not recovered with preference given to object form or extents. Another drawback is the failure to separate connected objects in low contrast regions [141]. Semantic segmentation results are often useful for recovering building footprints, which result in LOD 0 or LOD 1 building models. To recover object form, semantic segmentation results are often fused with other features, such as edge detectors, point clouds, or digital surface models in an attempt to recover the building boundary shape [97]. A detailed discussion of building object detection is given in Section 2.5.2.

Figure 10: Sample results from deep learning presented by Li et al. [149] showing how object extents are detected. First column is the input image, second column represents ground truth. Green in the third column are correctly segmented building and red shows wrong detections.

2.2.5 Building Reconstruction

Building reconstruction involves recovering the true object boundaries either in 2D or 3D. Using results from building object detection, a boundary recovery method is required to reconstruct the true boundary shape of the identified building objects. This is often achieved through the use of points, edge features, and radiometric properties of the building [67, 59, 92]. By connecting edges and finding closed boundaries based on features, radiometric properties, or coarse object boundary from segmentation, building boundary shapes are recovered. Deep learning methods have recently been proposed to recover accurate boundaries from semantic segmentation results [66, 65, 141]. The recovered boundaries are rendered in 2D if height information is not available or 3D if a height model is available. However, a not very popular approach is attempting to match edges of the same object in 2D multiview scenes and reconstructing the 3D object. Edges are usually limited to support features during 3D reconstruction because of fragmentation and missing edges. Improving this shortcoming is part of this research. Various boundary recovery methods have been proposed and a detailed discussion on object detection is given in Section 2.5.3. Example boundary recovery results are shown in Figure 11 and Figure 12. For most methods that use satellite data as input, the outputs are usually block models or simplified LOD 1 buildings.

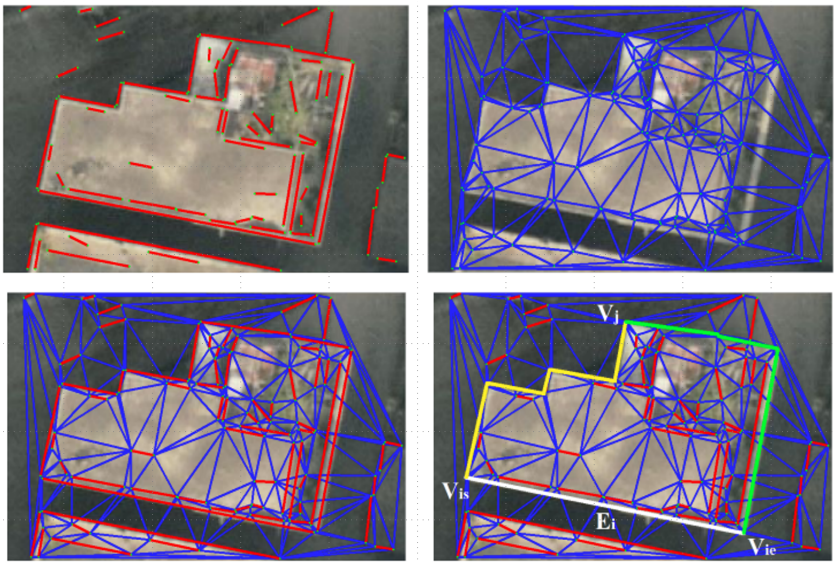


Figure 11: Building reconstruction example based on edges. The top left image shows detected edges, the top right and bottom left show the process of completing the building boundary. The bottom right image shows the final building edges, highlighted in green, white and yellow [67]

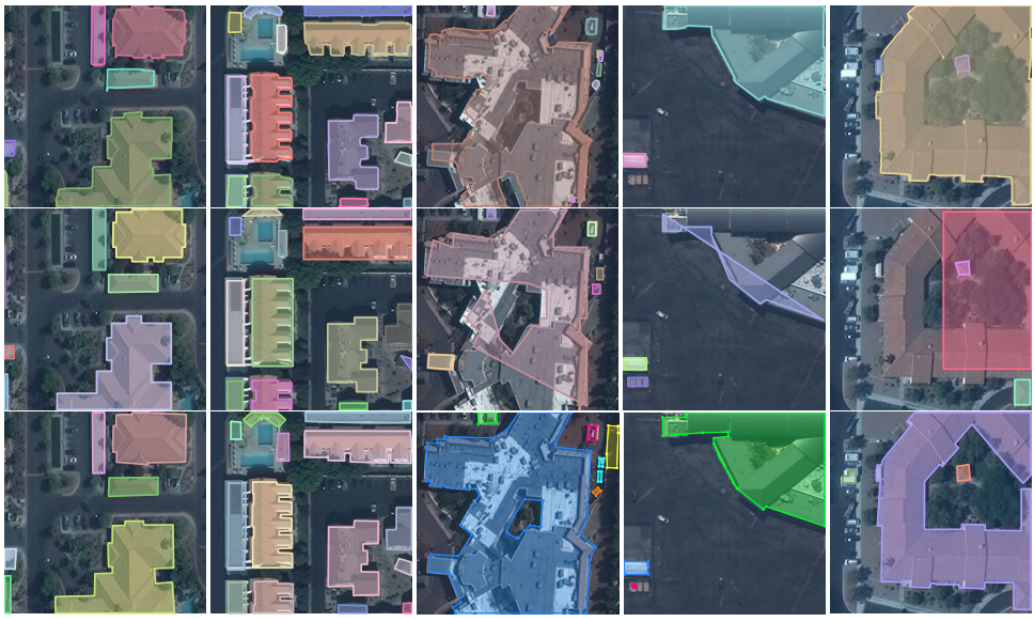


Figure 12: Sample reconstructed footprints from deep learning reported by [66]. Three different results from different methods. Top row: U-Net variant [153, 96], middle row: PolyMapper [154], and bottom presented by Girard et al. [66]

Figure 13 shows the footprints reconstructed from semantic segmentation based on deep learning [66, 65]. The final rendered B-REP model for each building is achieved by assigning DSM heights to the reconstructed footprints.

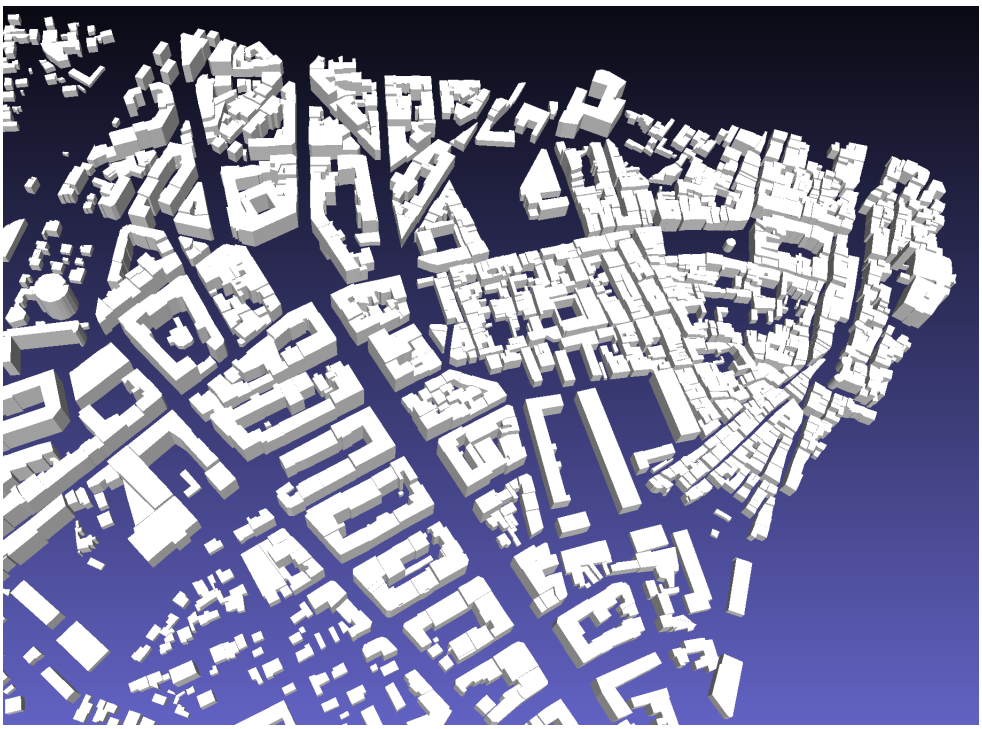


Figure 13: Sample 3D buildings reconstructed from footprints for Antibes, France. Model courtesy of Luxcarta [103].

2.3 Object Form and Detail

In this research, the goal is to recover both object form and object detail. Object form refers to the building outline shape or boundary shape, as projected on the image, whereas object detail refers to object superstructures, namely chimneys or stacked structures, and facade detail such as windows and doors. The object detail referred to is what constitutes LOD 2 and LOD 3 detail. Figure 14 gives a representation of the difference between object form and object detail. Using the buildings in Figure 14, the form of the object is yellow and the detail of the object is blue.

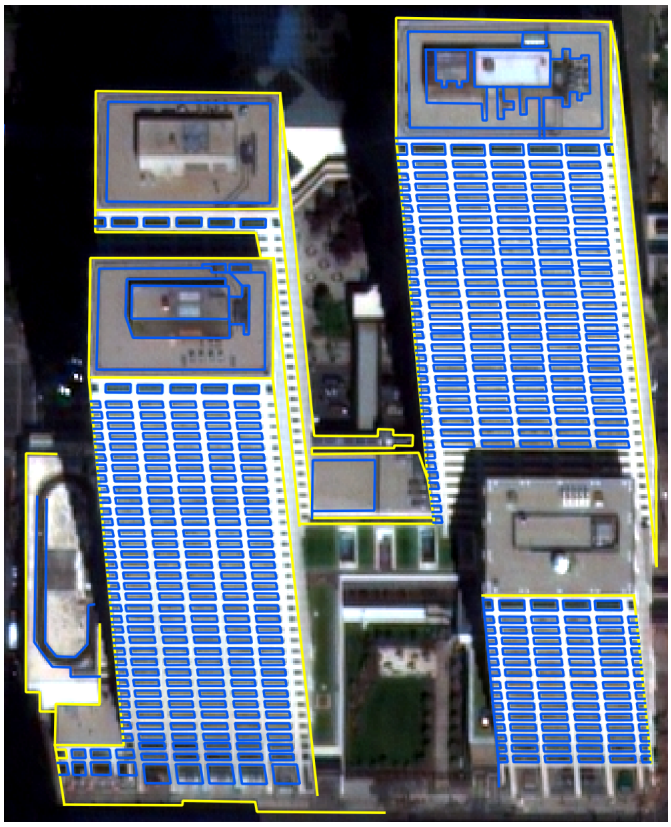


Figure 14: Object form is shown by the yellow boundary whereas detail is shown in blue. Most object detection methods focus on recovering object form.

2.4 Gestalt Principles

Before a detailed description of the framework developed in this research is given, it is important to understand the arrangement of objects in the real world and how they are perceived. This is useful when recovering closed boundaries of buildings, specifically in cases where the recovered boundaries are incomplete. By analysing edge associations

using Gestalt principles, boundaries can be completed by connecting edges. The main theory behind gestalt theory is that objects are perceived not as individual components, but as complete patterns formed by the components as a whole [155]. Human object identification abilities can be described by Gestalt principles, namely proximity, similarity, figure-ground, continuity, closure, and connection. The ability to discern objects in relation to other objects and the environment is based on the identification of patterns and arrangements. This allows the differentiation and identification of objects in different environments [155, 156].

The six main principles are:

1. The principle of Proximity states that objects close together are assumed to form a group. This is crucial in identifying repetitive patterns formed by objects.

2. The principle of similarity states that similar objects are generally grouped together. Similarity mainly covers features within features, such as windows on high-rise buildings and adjacent similar features from buildings that resemble each other.

3. The principle of symmetry states that we perceive objects as being symmetrical around a centre point. Consequently, disjoint parts can be joined by the mind to form complete structures.

4. The principle of continuity states that parts can form the whole if objects follow known patterns of alignment, such as circular or rectangular. The human eye tends to follow the simpler path, and not the complex one. Using these principles, lines and curves can be grouped together to form patterns that describe the structure of objects.

5. Closure is the ability to identify closed shapes even with fragmented boundaries. If sufficient boundary fragments exist, we can identify the missing parts so as to identify the whole. Key edge arrangements can thus be created using gestalt principles.

6. Figure/Ground principle states that objects tend to be separated based on foreground and background. This is useful in identifying occluded regions.

2.5 Building Reconstruction Strategies

2.5.1 Digital Surface Models from Image Matching

3D information from multi-view imagery is derived from image correlation methods. The reason for the use of DSMs is to correct for the projective distortion of objects that characterises multiview images. During the correlation, corresponding pixels from overlapping images are found and used to compute 3D positions. By matching pixels from a high resolution image, dense 2.5D point clouds or DSMs are obtained. The Point clouds and DSMs produced are then used to detect and reconstruct objects. DSMs that attempt to preserve sharp edges and breaklines produce more desirable object detection results. Consequently, efforts in this research area have been to preserve sharp object edges and breaklines in resulting DSMs. Therefore, the focus of this review is on methods that aim to preserve edges and breakline information when producing DSMs.

Image matching techniques can generally be classified into two categories, namely, methods that exploit epipolar geometry and methods that are projection-based. Epipolar-based methods begin by transforming images to epipolar geometry [133, 134, 135, 136]. In epipolar geometry, the pixel correspondence search problem is reduced to a one-dimensional problem since y parallax is eliminated. Projection-based matching algorithms use projective geometry of cameras to search for corresponding pixels [48, 127, 157, 158, 159].

Considerable research has been done for both categories of image matching algorithms. Starting with epipolar-based methods, Krauss et al. [137] presents an algorithm for reconstructing sharp vertical features from stereo images using epipolar line matching. After reprojecting images to epipolar geometry, advantage is taken from the epipolar property that parallax exists only in the horizontal direction. Corresponding pixel line segments are matched in the x-axis direction using the dynamic time-warping concept. The grey value sequences from the corresponding epipolar lines are matched based on a minimum grey value distance measure. Each pixel is assigned a potential match resulting in a pixel-by-pixel match. Establishing correspondence on a pixel basis allows better reconstruction of breakpoints at object boundaries. However, because the pixel matching in Krauss et al. [137], is restricted to

one dimensional lines, this often leads to streaking effects since information from neighbouring pixels is not taken into account. Consequently, the resulting disparity map requires filtering out mismatches and smoothing out the streaking effects. The streaking effect adversely affects the shape of the resulting objects even after post-processing [132].

In efforts to counter line streaking, Hirschmuller [57] presents a robust pixel-to-pixel matching method,where matching is pixel-based rather than window-based. This is one of the widely used pixel correlation methods termed Semi-Global Matching (SGM). Window-based approaches often assume a constant disparity in a given window. However, such an assumption is invalid in the presence of discontinuities and occlusions. Window-based methods, therefore, do not preserve edge information very well. SGM method solves this by matching pixels according to their individual intensities. The matching cost for establishing correspondences is based on mutual information that is not affected by radiometric differences in images. Instead of restricting matching to a single scan line, matching costs are aggregated along different directions that converge at the candidate pixel. The lowest cost disparity is chosen. To avoid mismatches, a smoothness constraint is introduced that supports smoothness and penalises sharp disparity changes of pixel neighbours, thereby preserving discontinuities introduced by object boundaries. Variants of this algorithm have been proposed [56, 63] with Bethmann and Luhmann [140], proposing a multiview, semi-global matching algorithm in object space rather than in pixel space. In addition to matching images in object space, a multiview approach is adopted as opposed to pairwise matching, which requires aggregating the results if more than two images are used. This eliminates the need to process disparity maps. Rothermel et al. [139], presents a hierarchical pyramid matching SGM where the results of the matching at each lower pyramid level are used to limit the disparity search range for subsequent pyramid levels. Although SGM and its variants are known to perform, shape inference is generally attempted as a post-process. Another example is the SGM variant presented by D'Angelo and Kuschk [160]. Based on this algorithm by D'Angelo and Kuschk [160], Reinartz et al. [64] presents a 3D building extraction pipeline from DSMs. The method employs a modified SGM algorithm that reduces the search space of pixels in smooth neighbourhoods and consequently reduces the overall run time of the original SGM algorithm. Reinartz

et al. [64], reports denser DSM results from this modified SGM variant. The denser DSM has a positive impact on the estimation of building boundary shapes. After DSM production, a geodesic dilation approach is used to separate terrain from non-terrain points. The result is a Normalised Digital Surface Model (nDSM). From the resulting nDSM, the building boundary shapes are reconstructed by estimating ridge lines that are subsequently used to fit predefined parametric shape models. This method selects the best fitting primitive to the estimated ridge lines. However, this limits the range of structures that can ultimately be reconstructed. SGM has proven to be a robust algorithm [64]. However, other matching algorithms that perform well have been proposed to match satellite imagery [161, 138, 48, 73, 162, 163].

The key to successful shape recovery from DSM is the preservation of sharp discontinuities in the output point clouds. Gupta et al. [53], present an image matching approach aimed at preserving sharp image discontinuities. A region-growing approach is proposed starting from seed points that are obtained in an initial feature detection step using a Harris Corner Detector [124]. After Harris corner detection on normalised images, corresponding features (i.e. corners) in the second image are identified. Cross correlation is used as a measure to identify false positives. High cross correlation is used in selecting the correct corner matches. To detect outliers in the feature matching step, Random Sample Consensus (RANSAC) is used. Using the seed points, the regions around the seed points are grown using a disparity component growth algorithm. Since regions are grown on the assumption of constant or minimal disparity changes, sharp edges are preserved. However, the results presented are from aerial images that have a higher resolution and no shape recovery is presented.

Some methods do not require rectified images as input but instead leverage the imaging geometry, where the image-to-object relationship is used in defining search spaces for pixel correlation. Zhang and Gruen [48], presents a DSM generating algorithm based on the combination of multiple matching algorithms in a coarse to fine hierarchical model. Multi-view images are utilised to counter the effects brought by occlusions, shadows, and matching ambiguities, which often occur when utilising a single stereo pair. Firstly, the images are pre-processed to enhance features such as edges and corners while reducing shadow effects. Textures are then enhanced by using a Wallis filter. An image pyramid is generated on the enhanced imagery from low

resolution up to full resolution. Image matching using Multiple Primitive Multi-Image Matching (MPM) is performed at each pyramid level. A combination of point and edge matching strategy is used, in a relaxation based relational matching procedure. From the results of each pyramid level, a constrained Delaunay triangulation is constructed from the point and feature matches. The matches are subsequently used to select and fine-tune the matching parameters for the next pyramid level. This implies continuous mesh refinement at each pyramid level until the highest resolution is reached. The method's strengths is the incorporation of edge matching that is used in preserving surface and terrain discontinuities. Consequently, the resulting mesh captures prominent discontinuities. However, similar to many DSM algorithms, even though extra feature information is used, the result is an amalgamated mesh with no object detection or shape detection attempted. A detailed description of the method can be found in [48, 127]. Variants of this approach can be found in [157, 158, 159]. Frederik Tack and Buyuksalih [159], presents a similar method which, as a post-process, uses 2D building footprints to extract buildings from the obtained DSM. Other approaches that employ a coarse to fine hierarchical approach in image matching can be found in [162, 164].

Zhang and Fraser [73], proposed an algorithm that uses both radiometric and geometric information contained in the input images. The proposed algorithm employs a coarse-to-fine hierarchical strategy where an image pyramid of reduced image resolutions is utilised. Matching begins at the top of the pyramid with the lowest resolution image where the pixel correspondences established are used as input in the matching process of the subsequent pyramid level. Matching cascades down the pyramid, with the DSM getting denser at each level, until the lowest pyramid level is reached. To further constrain the search space and improve the DSM quality, an initial DSM is similarly generated from features points at each pyramid level, serving as input to the subsequent lower level. Feature points, edges, and grid points are used at each level progressively down the pyramid. A two-step matching procedure is performed at each level where cross correlation is combined with structural matching using the input DSMs. A combination of these results provides a dense DSM that preserves discontinuities since edge position and attributes of the edges are used as a constraint at every pyramid level. While an effort is made to preserve discontinuities by incorporating

feature information, accuracy is not placed on the resulting object shape, but rather on preserving break line information. Edge attributes are assigned to single edges without any relationship to the shape or orientation of the object. This lack of semantic information adversely affects object detection. Another similar method to Zhang and Gruen [48], is presented by Wolff and Gruen [162], where the goal was to improve height accuracy by using an image triplet rather than a stereo pair. No evaluations on image object shape were conducted. However, better structural pattern retrieval was evident in the city area when image triplets were used. This is evidence that the more multiview scenes available for an area, the better the quality of the matching and the resulting DSM. A good DSM ultimately improves the shapes of recovered objects.

Some methods are purely feature-based, where matching is based on feature detected. In Alwan and Naji [165], an edge matching algorithm is presented that extracts linear edges from stereo images and matches the lines based on length, orientation, and edge end-points. The method consists of, first, an edge detection process in stereo images that is followed by edge linking to create unbroken edges. These edges are then consequently thinned, and straight lines are extracted. Straight line matching from stereo mates is carried out based on line orientation, end-point coordinates, and length. 3D information is then extracted using the matched line pairs. The major problem of feature based matching and reconstruction has been the inability to retrieve dense features that model the objects fully. For example, edge detection suffers from fragmentation, thus hindering the detection of full length edge features. In addition, linear segments limit the number of edge features that can be recovered and matched.

For the aforementioned algorithms, one recommendation has been to use stereo images acquired in a short period of time to reduce the effects of radiometric variability. However, recent studies have also shown the ability to extract DSMs from fortuitous stereo images collected over a wide date range. Facciolo et al. [50], and Qin [24], present results from the public benchmark dataset for multiview stereo mapping using satellite images (IARPA MVS Challenge Dataset) [163] from a wide date range. A comparison is made between fortuitous multiview and true multiview stereo with very useful results. Criteria for selecting multi-date stereo images are also outlined with good point clouds results presented. Studies presented in Facciolo et al. [50] and Qin [24] show the viability of fortuitous stereo for DSM generation and a DSM fusion method for merging

DSMs from multiple stereoscopic pair outputs in an effort to enhance DSMs. Therefore, the use of multiple date scenes increases the number of multi-view scenes that are used in DSM generation. Since DSMs are derived products, they do not contain rich feature information compared to input images. Generally, sharp discontinuities are not entirely preserved. Consequently, less features can be extracted from DSMs. DSMs are thus often used in combination with source imagery that provides the required feature information for detection. The various methods for this task are discussed in Section 2.5.2.

2.5.1.1 Semi-Global Matching The SGM algorithm is discussed here for completeness since it is the mostly widely used image matching algorithm. SGM, which produces dense point clouds, is well known for its robustness. Most algorithms consequently use SGM in image matching. Most researches have built upon the original SGM in an effort to improve efficiency. Figure 15 below gives a pictorial depiction of SGM derivatives that have been presented. Contrary to traditional window-based matching algorithms which results in reduced resolution output, SGM matches up to pixel level thereby achieving dense point clouds. The strength of SGM lies in the aggregation step of the matching mutual information cost that considers multiple One Dimension (1D) directions.

Semi Global matching uses a pixelwise cost function with smoothness constraint that is defined by a global energy function shown by equation 2.5.1 below

$$E(D) = \sum_{p} C(p, D_p) + \sum_{q \in N_p} P_1 T[|D_p - D_q| = 1] + \sum_{q \in N_p} P_2 T[|D_p - D_q| > 1] \qquad (2.5.1)$$

The term $\sum_p C(p, D_p)$ sums up all pixel matching costs for disparities of the image D while the second term of the equation adds a penalty P_1 for small disparity changes in the neighbourhood of a candidate pixel. A larger penalty P_2 is added for large disparity changes, thus preserving discontinuities. p is the candidate pixel and q is the candidate match. SGM is a derivative of scanline optimisation [137]. Resultantly, the cost function for minimising $E(D)$ is defined in 1D by equation 2.5.2 since minimisation

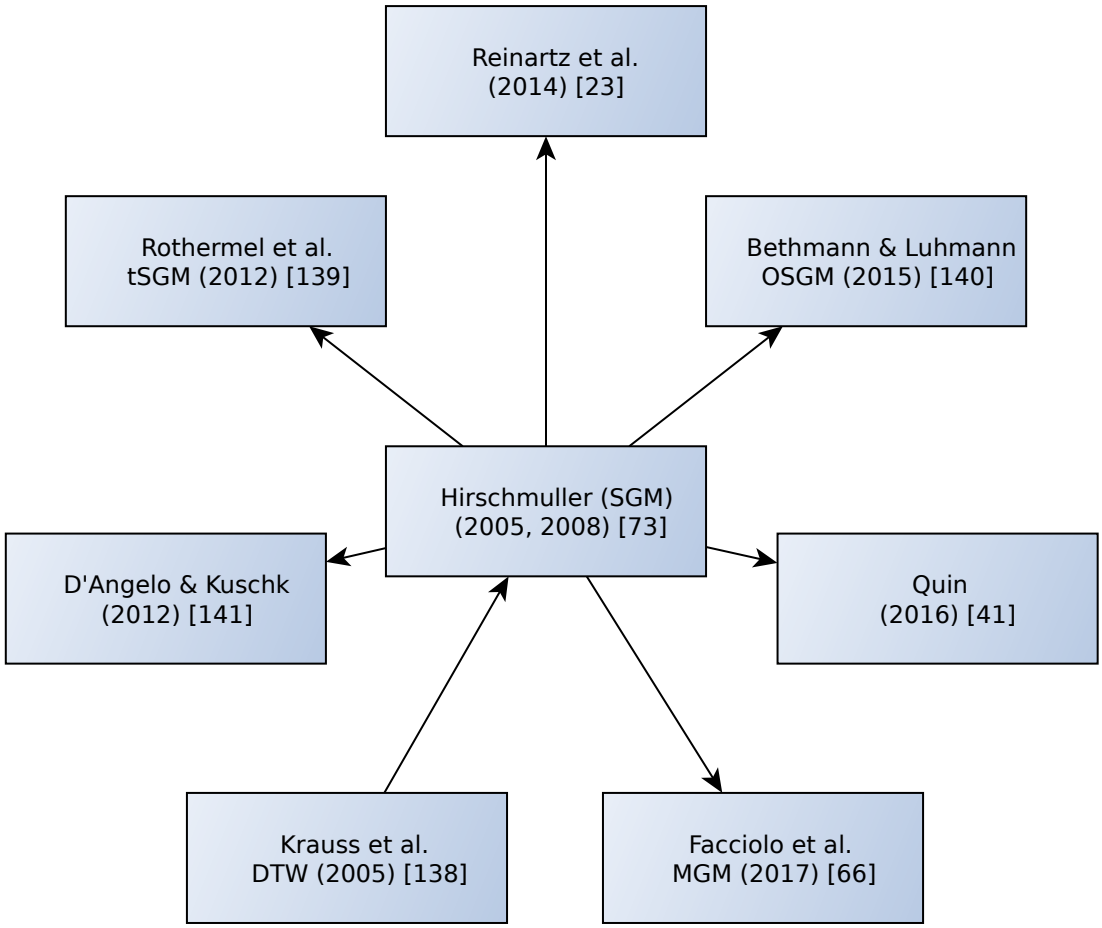


Figure 15: Semi-Global Matching algorithm proposed by Hirschmuller [57], and proposed derivatives from various authors

can be performed efficiently in 1D.

$$\begin{aligned} L_r(p,d)) = C(p,d) + min(L_r(p-r,d), \\ L_r(p-r,d-1) + P_1, \\ L_r(p-r,d+1) + P_1, \\ {}^{min}_{i} L_r(p-r,i) + P_2) - {}^{min}_{k} L_r(p-r,k) \end{aligned} \tag{2.5.2}$$

To avoid streaking effects caused by 1D minimisation, multiple paths of different orientations are defined that end at the candidate pixel, and the cost is calculated over these paths. Finally, the cost for a pixel p and disparity d is given by equation 2.5.3.

$$S(p,d) = \sum_{r=1}^{8,16} L_r(p,d) \tag{2.5.3}$$

where the true disparity is given by $min_d S[p,d]$. A detailed description of SGM can be found in [132, 57]. As illustrated in Figure 15, SGM has become the preferred image matching algorithm for 3D reconstruction pipelines. Consequently, this has resulted in SGM variants proposed in literature. Since mutual information cost requires a priori disparity information, [57] uses a hierarchical approach to estimate initial disparities using low resolution image pyramid. Initial disparity estimates for each level are used to refine the matching cost for the next pyramid level. Rothermel et al. [139], improves the memory requirements and runtime of SGM by further using the initial disparities to limit the search range of the pyramid levels.

One requirement for SGM is the provision of rectified images for matching. This means that matching is only possible in pairs. Extending SGM to multi-image matching requires fusing matching results from multiple pairs [57, 139, 50]. To allow simultaneous multi-image matching [140] proposes SGM in object space termed Object Based Multi-Image Semi-Global Matching (OSGM). The main variations in the proposed algorithm are that the matching cost and path-wise minimisation calculation are executed in object space. In addition, input image rectification requirements are not mandatory. Non-rectified images can be matched simultaneously. This results in equation 2.5.1 taking the form shown in 2.5.4

$$E(Z) = \sum_{X,Y} C(X,Y,Z) + \sum_{q \in N_p} P_1 \cdot T[|Z - Z_p| = \Delta Z] + \sum_{q \in N_p} P_2 \cdot T[|Z - Z_q| > \Delta Z] \tag{2.5.4}$$

For the minimisation of equation 2.5.4, the cost function in equation 2.5.2 becomes

$$
\begin{aligned}
L_r(v, Z)) = C(v, Z) + min(L_r(v - r, Z), \\
L_r(V - r, Z - \Delta Z) + P_1, \\
L_r(v - r, Z + \Delta Z) + P_1, \\
{}^{min}_{i} L_r(v - r, i{\cdot}\Delta Z) + P_2) - {}^{min}_{k} L_r(v - r, k{\cdot}\Delta Z)
\end{aligned}
\tag{2.5.5}
$$

The major advantages of the proposed algorithm are that rectified images are not a prerequisite. Multi-image matching can be performed simultaneously with more than two images. The matching directly outputs coordinates in object space, which simplifies consistency checks, and the matching can be done in a parallel manner for efficiency. Finally, another variation of SGM is proposed by [56] termed More Global Matching (MGM). In the proposed algorithm, significant improvements are suggested that reduce the streaking effects associated with SGM as a result of the direction of the linear paths. Facciolo et al. [56], proposes updating the cost of a given path L_r by using information from more than one direction. Traditionally in SGM, L_r is calculated from information using one direction. Pixels above a candidate are further incorporated into the calculation of L_r, in addition to the pixel before. Progressively, the cost for any given pixel is influenced by the entire upper left quadrant. This leads to a modified cost function shown in equation 2.5.6

$$
L_r(p, d) = C(p, d) + \sum_{x \in r, r^{\perp}} \frac{1}{2} {}^{min}_{d' \in D}(L_r(p - x, d') + V(d, d')) \tag{2.5.6}
$$

The term $r^{\perp}$ indicates the direction of the pixel above the candidate pixel p. This is an improvement to SGM which only incorporates information from the left (i.e. $p - r$). Other studies have focused more on evaluating the performance of different cost functions [24, 160] or evaluating SGM performance in 3D reconstruction [64]. The extensive effort to improve SGM outlines its significance for image matching, which is the basis of most reconstruction efforts. This research aims to lay a basis for edge-based matching for

building reconstruction that can provide alternatives to exhaustive pixel matching.

2.5.1.2 Three-Dimensional Point Determination To compute 3D point information from an image, a defined sensor model is required that describes the relationship between the pixels and the ground coordinates. With satellite images, this is achieved by using the RPC sensor model discussed in Section 2.2.2.1. Before any 3D point determination, corrections for imaging errors are required. The error models explained in Section 2.2.2.2 are modelled with the RPCs to improve ground point accuracy. As discussed by Fraser et al. [111], since the RPC model can be equated to a re-parameterisation of the physical sensor model, inherent errors in the physical model can be modelled similarly. With bias compensation, the RPC equations become:

$$\begin{aligned} l_N &= \frac{P_1(\phi_N, \lambda_N, H_N)}{P_2(\phi_N, \lambda_N, H_N)} + a_0 + a_1.sample + a_2.line \\ s_N &= \frac{P_3(\phi_N, \lambda_N, H_N)}{P_4(\phi_N, \lambda_N, H_N)} + b_0 + b_1.sample + b_2.line \end{aligned} \tag{2.5.7}$$

The eighty RPC coefficients are created from ground coordinates generated from the physical sensor model and in most cases without GCPs. Early concerns about RPC adoption was the lack of any physical meaning of the eighty coefficients and methods to adjust for inherent errors in the coefficients derived without proper GCPs. However, as presented by various authors, the RPCs achieve comparable accuracy to physical sensor models, and inherent errors can be corrected using equation 2.5.7. Equation 2.5.7 is used in a least-squares approach to solve for any point in 3D given two or more matching pixel points. To calculate the position of an edge in 3D, at least two corresponding points on the matched edges are required.

2.5.2 Building Object Detection

Prerequisite to most building reconstruction algorithms is building object detection. Building object detection is an area that has enjoyed active research over the decades. Automated building detection algorithms receive as input, either DSMs or images. The output of a detection process is often the location and extent of building objects

[43, 89, 64, 60, 100]. Sample building detection results are shown in Figure 16. With imagery as input, building object detection uses image features, namely corners, edges, and homogeneous regions based on radiometric measures, in identifying building structures. However, with DSMs as input, building detection is carried out using one of two approaches. In the first approach, buildings are detected by executing a filtering process that removes ground points leaving above-ground objects followed by building detection from the remaining points [166, 64, 43]. In the second approach, the DSM is augmented with features or radiometric measures derived from the input images, after which buildings are detected based on the combined features. This section reviews some proposed building object detection algorithms and the challenges that currently exist.

Figure 16: Object detection using satellite images [43]. Building object extents are shown in blue. often the result of filtering and object detection algorithms are object extents.

2.5.2.1 Learning Based Building Object Detection Recently, 3D wireframe extraction methods based on deep learning have been proposed. However, these methods are mainly focused on terrestrial images, as opposed to satellite imagery. The algorithms proposed aim to reconstruct 3D building wireframes by learning from manually labelled wireframes. For example, Zhou et al. [85] presents a deep learning approach that learns line and junction features. The junction features represents

building corners that are connected by the lines. The network is trained to detect two types of junctions. These are C-Junctions that occur at line and plane intersections and T-Junctions that represent the occurrence of occlusions. These detected lines and junctions are then used with estimated depth to reconstruct 3D wireframes. Zhou et al. [84] presents another algorithm that learns wireframe reconstruction from existing 3D CAD models of cities. First, the image panoramas are aligned with the CAD models. From the aligned panoramas, GT images are created. The GT images are then used to learn building wireframe reconstruction. Rather than generating image features directly from images, [83] uses generated 3D line features generated from multiview images to construct 3D building wireframes. The deep learning network learns to detect 3D junctions that represent building corners from a line cloud. Other methods that aim to reconstruct wireframes focus more on line detection from images with the aim of recovering the form of the building [8, 7, 9, 167, 168]. These methods focus mainly on high-resolution terrestrial images of buildings and indoor environments. Liu et al. [167], rather focuses on reconstruction wireframe from 3D point clouds as opposed to images. By predicting junctions and corners in a point cloud, and linking them using lines, a 3D wireframe is created. However, the model is tested on high resolution point clouds of single, small objects and not buildings. Qian et al. [168] focuses on reconstructing single-building roof structures of houses from ortho images and does not address the huge projective distortion in large area images such as satellite images.

There are several methods to detect building objects directly from satellite imagery. Orthophotos or nadir-looking images are preferred for building detection, since they have minimal projective distortions of structures. Most recently, building object detection has been tackled using deep learning techniques. Deep learning methods provide an object-based building reconstruction approach as opposed to surface reconstruction, which requires further processing to extract building objects. The release of Microsoft building footprints on an almost global scale is evidence of deep learning capabilities for detecting building objects [151, 152]. Deep Convolutional Neural Network (DCNN), in the presence of sufficient training data, has proven to outperform conventional computer vision and machine learning methods for object detection [97, 169, 170, 171, 172]. Most DL methods focus on achieving accurate segmentation results.The most popular DCNNs used for semantic building detection are

U-Net, LinkNet, and Mask R-CNN. Buyukdemircioglu et al. [97] uses both U-Net and LinkNet for the semantic segmentation of buildings from imagery. However, problems encountered with building detection methods still exist with Deep Learning (DL) methods. Repeated textures, low contrast variation, and intra-class variation within an object reduce detection accuracy. This has led to the incorporation of height maps as an additional source in the training set. For example, using ALS data allows leveraging height information at object boundaries when image contrast is limited [170, 173, 97, 174]. Furthermore, to avoid errors associated with the use of third-party data sources in the segmentation process or lack of ortho-rectified imagery that aligns with a given DSM, some methods seek to use stereo images to obtain height information. To improve results, nDSMs are often used as an additional input [142, 148]. Similarly, [170], introduces a height map fused with the input image bands, which results in an increase in F1-Scores. However, height data augmentation only improves segmentation results. As with most DL methods, the segmentation results allow delineation mainly of building footprints or building form. This is illustrated in Figure 17. Consequently, some methods aim to improve the resulting footprint shapes. For example, [169] presents an end-to-end pipeline focused on improving the resulting building vectors obtained from the deep learning segmentation results.

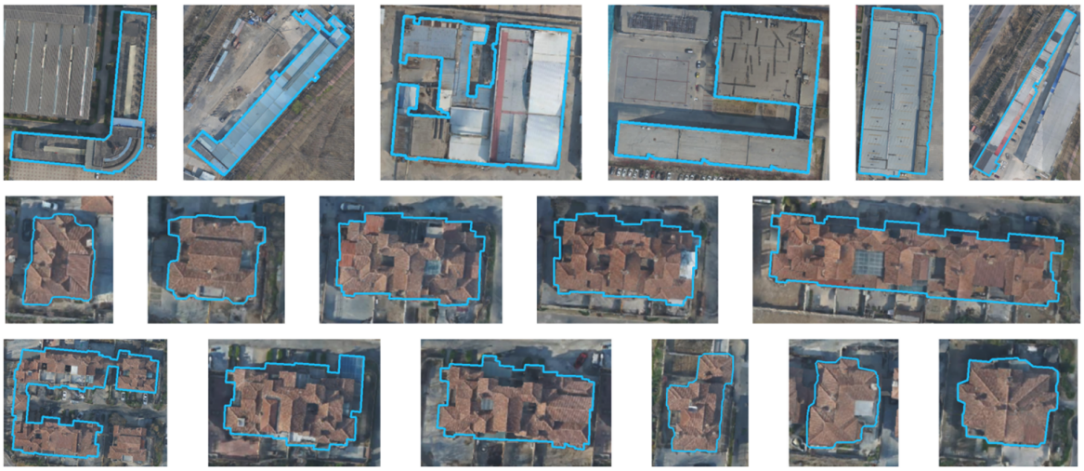


Figure 17: Building footprints derived from deep learning. Connected buildings are often detected as one structure, as shown in the images above, and only the footprint shape is recovered. These are some of the effects of under-segmentation with DL [175].

The challenge, with deep learning, is image scale and resolution. The ever increasing resolution of input imagery to semantic segmentation introduces various challenges. Firstly, the increased detail captured with higher resolution introduces greater

intra-class and inter-class variances which affects the performance of any existing trained model [97]. Consequently, there is a need to continuously re-train models at different resolutions to achieve acceptable segmentation results. Secondly, if the different building structures are not well represented in the training set, the probability of detecting these buildings is greatly reduced, resulting in partial detections [142]. The performance of DL methods depends on the training data. Thirdly, of concern to this research, the segmentation map generally returns a binary image with building versus non-building label. The binary map does not accurately represent the form of objects since this information is lost during discretisation. Object detail is typically not recovered. This limits the recovery of shape from deep learning to footprints, as shown in Figures 17 and 18. Moreover, if the image is not a true orthophoto, then the delimited boundary will result in a distorted and displaced footprint. Some methods focus on improving boundary shape recovery, but the shapes are limited to building footprints [169, 171]. Therefore, this limits the shape of recovered objects to LOD 1. Any errors in the segmentation process are generally transferred to the resulting footprint shape. Attempts to improve building shapes from DL segmentation are gaining popularity. The main objective is to recover LOD 2 detail from the detected footprint by predicting the roof types associated with the footprints. Detecting the type of roof, such as a gable or a hipped roof coupled with the footprint, LOD 2 models are obtained [176, 141]. Figure 19 shows some LOD 2 buildings obtained from roof predictions that have been extruded to the ground.

Often, the requirements of machine learning are some form of classifier training that uses known object classes and then detects these object classes in images. Inter-class variability is the main source of ambiguity with regards to object representation (i.e. many kinds of building, dome shaped, pyramid, circular, etc). Preparing training sets that cater for all primitive types is an extensive task. Some methods learn based on prior knowledge generated from architectural plans [177]. As mentioned in section 2.5.2.1, the trend with some deep learning methods is to incorporate depth information as part of the training data in creating detection models [170, 173, 97, 174]. With an associated footprint, deep learning methods that use depth information result in LOD 1 buildings. The lack of adaptability to satellite images without the need to retrain is a major drawback. In addition, the lack of publicly available training datasets [87, 86] for

Figure 18: Building footprints derived from deep learning [171]. Some of the footprints shapes do not match the building shape and orientation.

satellite images and the amount of training data required add to the challenges of deep learning [84, 85]. Most of the methods aim to reconstruct only building form at the expense of detail. Furthermore, the requirements for providing model parameters add to the challenge of deep learning models.

2.5.2.2 Area based Object Detection Area based methods aim to define objects based on some similarity measure. The main assumption is that an object exhibits similar radiometric or geometric properties. Most of the methods in this class are often driven by segmentation algorithms that cluster pixels that exhibit similar geometric or radiometric measures [77]. For each segment, various features such as size, area, compactness, and length are often attached to each segment and then used in a decision tree to merge segments that belong to the same object. Gavankar and Ghosh [145], presents an algorithm that uses the different reflective properties that characterise different objects. A top-hat transformation using morphological operations is used to separate dark regions from light regions. Firstly, the image is enhanced to improve contrast by using the white and black top-hat transform. From the enhanced image, k-means clustering with three classes is used to obtain a dark, bright, and intermediate class range. The centre value of the intermediate class is then used to create a dark and bright image. Each of the two images is then filtered for noise to improve the

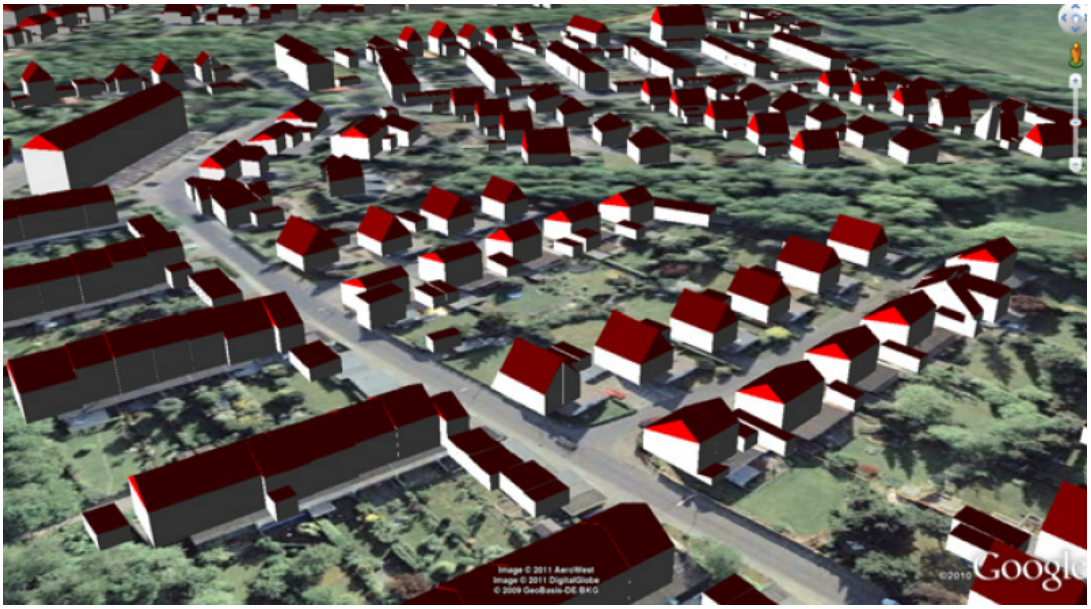


Figure 19: LOD 2 building detail recovered from deep learning by predicting building roof type [176]. Roof types are combined with footprints and a height to produce the final 3D buildings.

subsequent segmentation based on connected components. Since the resulting segments are a combination of building and non-building objects, a further step of removing non-building segments is achieved by segment feature thresholding. The features used to remove false detections are detailed in Gavankar and Ghosh [145]. The method shows high completeness and correctness values of over 0.8. However, the performance of the algorithms depends on the brightness or darkness of the objects. Consequently, any non-building objects that exhibit the same brightness and segment properties are miss-classified. In an effort to avoid challenges posed by the varying hierarchical structures of buildings, Abraham and M.Sasikumar [144] attempts to classify by elimination. Instead of trying to detect all buildings structures of different form and size, an attempt is made to detect all other objects in an image other than buildings, and eliminating them. Firstly, vegetated areas are detected using Normalised Difference Vegetation Index (NDVI). The vegetated areas are then removed, after which shadows are detected by converting the image to YIQ space and calculating the Q to I ratio, which is higher in shadow regions. Using Otsu's method [178], a threshold for eliminating shadow regions is identified and applied. Rooftops are subsequently detected using mean shift segmentation. Building edges are refined by hole filling and removing pseudo pixels. Road pixels are then identified from the remaining segments

using skeletonisation and removed from the image leaving only buildings. Further cleaning using area and segment size is used to refine the results. This is achieved by using Principal Component Analysis (PCA) for thin segment elimination and area-based labelling of connected components to remove small area segments.

Sirmacek and Unsalan [179], presents a building location detection algorithm from previously detected building regions. Features namely Harris Corners [180], gradient magnitude support regions, Gabor filtering, and features from accelerated segment test based local feature vector are used to model building locations as joint random variables. These features are used to estimate the probability density function whose modes and probabilities are used to identify building locations. The method does not delineate the building regions, but rather estimates the building locations in the image. In another variation, [181] detects built-up urban area boundaries as opposed to individual positions or individual building areas. This is achieved by using local feature point extraction through the use of Gabor filters. Gabor filtering is used to detect building edges. From these edges, local features are extracted. A voting strategy based on these features is then used to identify built-up urban area regions. By accumulating local features, which are generally clustered around building edges, higher votes are obtained around buildings. The limitations of the method is the separation of individual buildings in densely built-up areas versus sparsely built-up structures. In densely built-up regions, the result is the demarcation of built-up areas rather than individual building objects.

Yong and Huayi [147], present a building object detection method from DSMs. In this approach, the DSM is filtered using morphological filters. A morphological close and open operation is performed sequentially to remove above-ground features, thereby creating a Digital Terrain Model (DTM). However, since building edges do not appear as sharp after DSM generation, removing these buildings leaves a stair stepped effect which is corrected by applying a smoothing filter. An nDSM is obtained by subtracting the Digital Terrain Model (DTM) from DSM. From the nDSM, surface normal estimation is used to derive object normals, which are then used in a watershed-based segmentation approach to obtain the building objects boundaries.

2.5.2.3 Boundary based Object Detection Boundary based detectors often combine edge information derived from edge detectors such as Canny [182] or Sobel

[183] and then link them based on semantic linear edge relations and predefined shape constraints such as the rectangular nature of buildings. The basic step in most of these algorithms is the identification of linear building edges that are then linked together to form the building boundary [67]. Aamir et al. [184], presents a building detection model from low-contrast satellite images. One of the main limitations in object detection from satellite images is the lack of sufficient contrast that allows for inter-object and intra-object differentiation. As a pre-processing step, a Discrete Wavelet Transform (DWT) of the Fourier transform based on Singular Value Decomposition is used to enhance the contrast of an image while preserving edges. After contrast enhancement, a straight edge detection algorithm based on EDLines [3] is used to extract all linear edges from the image. From the obtained edges, perceptual grouping is performed to obtain building boundaries by linking linear edges based on ratio of overlap, euclidean interline distance, and angle between lines. The lines are linked to create closed contours representing building outlines. The assumption made is that any object has more than one linear edge and that objects are composed of right angled rectangular elements. Qin (2018), [67], presents an algorithm for boundary search based on a grouping cost. The algorithm works by first detecting lines using EDLines [3]. The detected lines are then used to construct a constrained Delaunay triangulation. From this triangulation, an undirected graph is constructed. Weights are assigned to the edges based on the length of the edge. All possible cycle candidates for the building boundary are identified, and for each, the grouping cost is calculated. The results presented are from true ortho images. The results presented are for individual buildings rather than densely constructed areas. The results are based on line detection. However, lines detected from images are characterised by fragmentation and missing lines. These are some of the problems addressed by the research presented.

Lin and Nevatia [80], present a method for reconstructing 3D shapes from a greyscale image. The algorithm relies on the projective and geometric properties of building shapes in 2D images. The algorithm presents a complete 3D reconstruction pipeline from a single image. By restricting building shapes to rectilinear shapes, shape hypotheses are generated from straight lines detected as a first step. Complex rectilinear shapes are assumed to be the result of union operations on rectangular shapes. Based on the fact that rectilinear shape projections result in parallelograms, the detected lines are used

to search for parallelograms. By identifying parallel anchor lines, candidate shapes are proposed based on local and global criteria that utilises supporting evidence such as the existence of corners, shadows, and polygon overlaps and containment to eliminate false hypotheses. The existence of any supporting evidence is used to add a weight to the proposed hypotheses. A further verification process is then applied based on the projective nature of buildings, where footprint and vertical wall evidence is sought using roof hypotheses and added weights. This evidence also provides the information required for height extraction. Thresholding is then applied using the calculated weights to select the final shapes [80]. The research presented here aims to improve boundary recovery by removing linear segment constraints by introducing curve detection.

2.5.2.4 Hybrid Methods for Object Detection Hybrid methods often combine the strengths of area and boundary based algorithms. The first step in most of these algorithms is building area delineation followed by boundary enhancement using auxiliary features such as edges and lines. Input data is not limited to a DSM or images but are a combination of multiple sources. When built-up regions are detected as the first step, spurious features are reduced. Hybrid methods tend to perform better in detecting building objects. Jianghong Song Yanfeng Wei [68], presents an algorithm for building extraction from a single panchromatic band. The algorithm begins with a clustering step that splits the input image into homogeneous pixel regions based on histogram peak selection. The purpose of this step is to detect and classify shadows, characterised by low grey values, and building objects. In addition, other classes are identified based on multiple histogram peaks. After detecting shadows, a building-shadow relation is established. Each identified building object is then used to extract the object boundary using Canny edge detection. This is followed by line detection using the Hough transform [185]. Hough transform is performed with Canny edges as input to detect the dominant lines of building objects. The dominant lines are then connected based on parallelism of lines and angle of convergence. However, the major drawback arises from segmentation based on histogram peaks, which at times results in over or under segmentation, especially on homogeneous surfaces or low contrast regions. In addition, small structures without shadows are not detected.

Another hybrid method is proposed by Chaudhuri et al. [107], which is a

combination of area and boundary based algorithms. In the proposed method, the input panchromatic image is enhanced to improve the contrast between objects while reducing noise within object boundaries. This is achieved by using directional morphological filters. Internal Grey Variance (IGV) features are used to enhance or reduce image brightness while blurring non-man-made features. From the resulting IGV features, image clustering is performed to separate man-made features. The separated features are finally segmented into the resulting buildings. From the results presented, the proposed method often fails to detect dark-coloured buildings. Multi-textured roofs are often not detected, resulting in partial building detection. Liu et al. [79], present an algorithm that reconstructs rectangular building shapes from satellite images. The algorithm begins by creating a multi-resolution segmentation of the pansharpened optical image. Object-based classification based on fuzzy rule decision tree is performed. The resulting binary building mask is then used to extract and refine building shapes using a progressive probabilistic Hough transform [5], which detects dominant linear building edges. Assuming rectilinear relationships between the lines, rectangular building shapes are fitted to the detected objects. The major limitation of this approach is the inability to detect non-rectilinear shapes, which characterise some building shapes in urban areas. Mean shift segmentation is employed in Jiang et al. [186]. However, instead of fully automated building detection, a semi-supervised approach is used where initial building regions are interactively selected by mouse clicks, and then building regions are grown based on similarity measures. The final boundaries are extracted using Canny edge detection. As previously mentioned, other features such as shadows are often used as indicators for the presence of buildings [68, 107, 80]. However, shadow presence is not predictable, which can be limiting for certain scenes.

Often, some area based methods attempt object detection using derived representation from the initial input images such as DSMs. As a pre-process when utilising DSMs, filtering is required [64, 47, 58], and then subsequently object detection and reconstruction [187]. An example that makes use of all available image information for reconstruction of city objects is presented by Jiang et al. [186]. An object-orientated approach is used, where the image is first segmented using a multi-resolution segmentation algorithm to extract building blocks. The homogeneity criteria used in segmentation are colour and shape properties. A DSM is then introduced into the process, where high features are

detected and separated from the DSM. Since these high features include vegetation, the green and red bands are then used as a ratio (similar to NDVI) to separate buildings from trees. Finally, the detected buildings from the DSM are merged with the extracted segments to obtain building height information. However, shape reconstruction is not attempted. Another approach is presented by Poli and Soille [188] which utilises both the DSM and imagery for automatic building detection using a hierarchical image approach.

KrauÃŸ et al. [51] presents a hybrid matching approach that uses multi-spectral images of the scene. The objective of the method is the extraction of objects from dense DSMs. In this method, epipolar images are generated after geometric corrections are applied [52, 104] followed by image pansharpening. From the pansharpened multi-spectral image, an (NDVI) map is produced. In the next step, a DSM is generated from the panchromatic images using semi-global matching [132, 57] and dynamic line warping method [137] described in Section 2.2.3. The resulting DSM from the matching is filtered to separate ground from non-ground points using median filtering, thus producing an nDSM. Using the nDSM, DTM and NDVI images, the scene is classified into five classes namely water bodies, ground, vegetation, buildings, and ground vegetation using the NDVI values [40]. By subtracting ground points from the disparity map created during matching, a disparity nDSM is created. Using NDVI and height thresholding on the nDSM, trees are removed leaving objects assumed to be buildings. The shapes of the buildings are then extracted. This algorithm is a good approach in that it uses multi-spectral information about the scene. This results in a classified DSM, which is usually not the result of most DSM generation methods. However, the shape outline of the objects still need to be reconstructed as a post-process. Since object information is lost during the matching process, any reconstruction of building boundary shapes is an approximation using the detected objects from nDSM. Furthermore, low-lying buildings have a higher probability of being eliminated during tree height thresholding.

The various challenges faced in object detection have been discussed in this section. The results of object detection require further processing to extract building boundary shapes. The building boundary shape reconstruction and how some methods attempt to address the challenges mentioned in this Section are discussed in the next section.

2.5.3 Building Boundary Shape Reconstruction

Object shape recovery from DSMs or images is mainly focused on the reconstruction of building boundary shapes. This section discusses methods that specifically target building boundary shape recovery. This process takes as input, building objects detected with methods in the previous section and reconstructs the final 2D or 3D boundary shapes. Figure 20 is an example of a binary map produced from semantic segmentation that is used as input to building boundary shape reconstruction. McClune et al. [41] and Wang et al. [42] classify reconstruction methods as model driven, data driven, or hybrid models, which combine model driven and data driven approaches. Model driven reconstruction approaches work by defining a set of parameterised models that are fit to the objects that are detected from input DSMs or images. The models allow for the representation or definition of common building footprint or roof shape boundaries. Data driven approaches aim to estimate shape based on the structure of the input data. Firstly, data driven boundary reconstruction methods are discussed followed by model driven reconstruction methods.

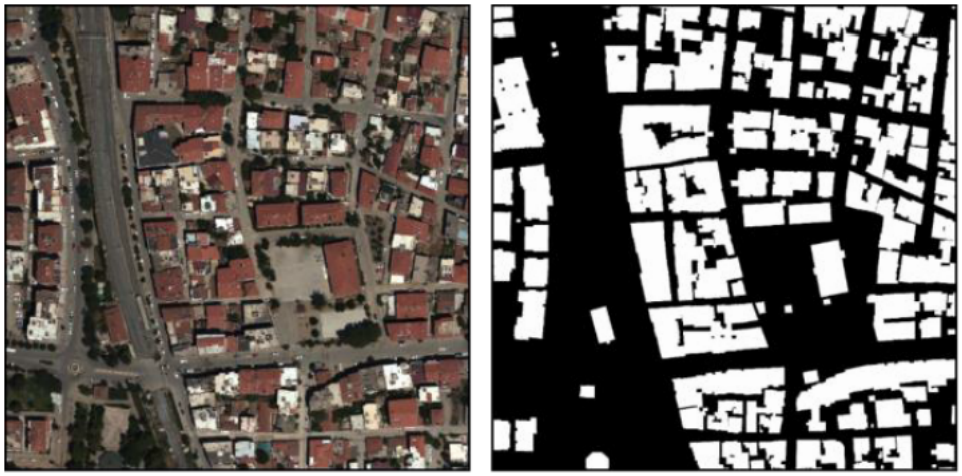


Figure 20: Semantic segmentation result detect object form at the expense of object detail. The results of the input image on the left are object extents that are used to recover footprint shape [97].

2.5.3.1 Data Driven Boundary Reconstruction Most data driven algorithms aim to recover building form and detail by preserving interior object detail or superstructures [130, 88, 92]. These methods present more sophisticated algorithms that perform better with respect to shape retrieval. For example, Duan and Lafarge [92] uses a stereo pair to

estimate the shape of the building by performing a spatial decomposition of the image space into convex regions. The convex partitions are restricted to follow edges where they exist, thus preserving sharp discontinuities. The heights of the convex partitions are estimated from a DSM obtained from matching. Using the height estimates, the 3D planar positions of the convex polygons are calculated. An energy function is then used to classify, cluster polygons and recover exact heights of the polygon clusters. From the recovered 3D clusters, a fusion process is performed to produce the final 3D model by extruding their outer boundaries to the ground. Since the algorithm assumes the presence of two classes, namely horizontal planar roofs and non-roofs, reconstructing complex non-planar structures is not easily achieved. Height information is a requirement in determining the final shape clusters. A similar improved approach is presented by Li et al. [96] where the image is decomposed into convex polygons with linear constraints.

Data driven methods perform best with high quality input data such as LiDAR point clouds or aerial imagery since data can be acquired at very high resolution. Shape reconstruction is more accurate with higher level of detail attainable [130, 88]. Roof detail is better captured with LiDAR thus allowing for better reconstruction. Resultantly, imagery that can produce point cloud resolutions comparable to LiDAR tend to employ LiDAR based reconstruction algorithms. Typically, most point cloud based methods begin by segmenting the point set to retrieve individual buildings after which plane detection is executed on each building. After detecting the planar regions of a given building, plane intersection is used in an optimisation step to retrieve the final building model [88, 189, 90]. Figure 21, and Figure 22 are examples of data driven methods.

Another example of other data driven approaches can be found in Demir and Baltsavias [60], Grigillo and Kanjir [100]. The use of multi-spectral or ancillary data for aiding in object detection has been a solution in separating man-made from natural objects [40, 60, 137]. These methods depend on the availability of multi-spectral bands or data such as Open Street Maps [190] to complement available image bands. However, the main use of external data sources is to separate objects rather than to define boundary shapes.

Similar approaches can be found in Grigillo and Kanjir [100] where urban objects are obtained from a DSM using an object based classification approach utilising a combination

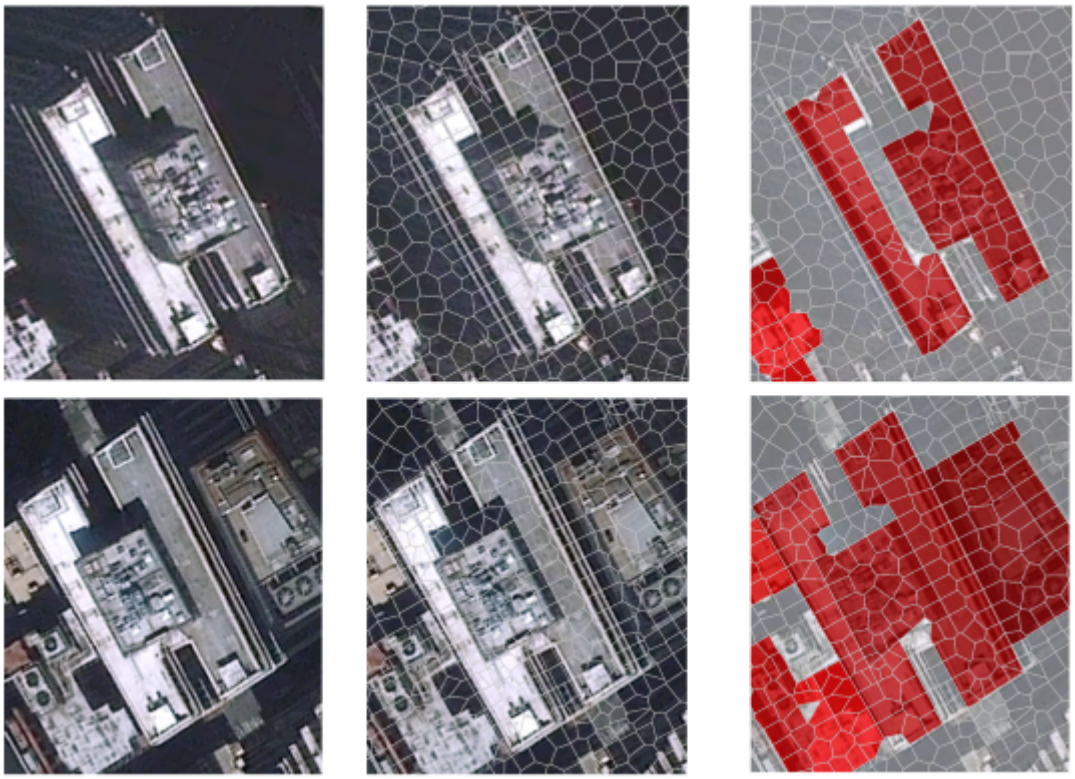


Figure 21: Image decomposition into convex polygons with linear edge following constraint. Convex polygons share partial edges with linear boundaries to preserve shape as shown by in the second and third column of the input buildings in the first column [92]

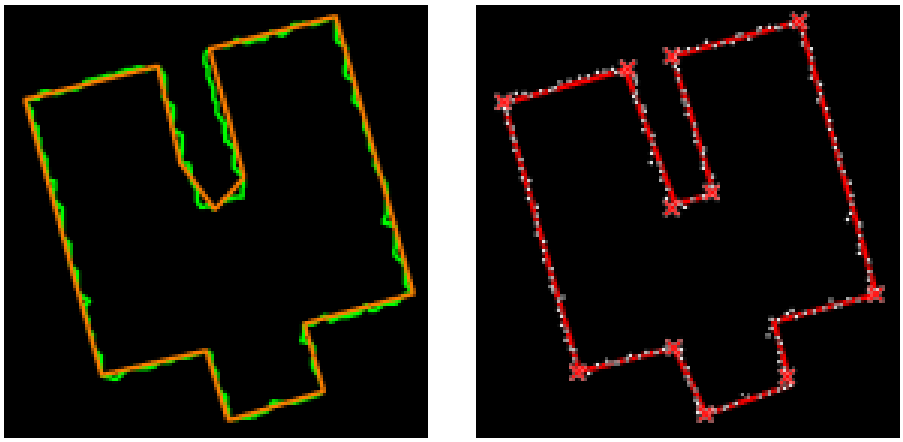


Figure 22: Data driven shape reconstruction from Widyaningrum et al. [59]. The identified boundary points are used to estimate the shape. The final shape depends on the quality of the input data.

of aerial images and point cloud data. Of interest is the inclusion of a building shape recovery process in the urban object identification process. The first step is to establish the main building axis orientation. Using the obtained orientation axis, lines parallel to the orientation axis along the building outlines are detected, including perpendicular

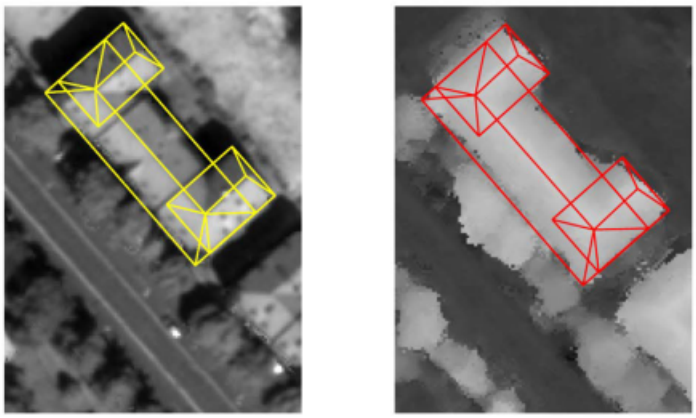


Figure 23: Model driven shape reconstruction presented by Reinartz et al. [64]. Parametric roof structures are selected and combined to create a complete 3D model.

lines. Based on the assumption of rectangular building shapes, the lines are intersected to create rectangles. From these rectangles, the final building shapes are derived. However, this assumption does not hold for all urban area buildings since all shapes cannot be approximated by generalised rectangular outlines.

More recently, semantic segmentation using deep learning has been used to detect building objects. Since semantic segmentation presents mainly results of object form, shape recovery is limited to footprints [141, 171, 142, 46]. Most approaches in this category are classified under data driven shape reconstruction methods. Consequently, the resulting shape depends on the quality of the input data. For example, Buyukdemircioglu et al. [97] and Chawda et al. [171] use the simple Douglas-Peucker algorithm for line simplification to estimate the footprint boundary [191, 192] from the segmentation results. However, the roughness and jagged nature of the segment edges reduce the accuracy of the recovered shapes. A more robust approach is presented by Bauchet et al. [65], where an optimisation technique is used, using as input, predicted dominant orientations along a building edge. By creating a graph of the input buildings vectors, and simplifying the boundaries based on the predicted dominant orientations, rooftop boundaries are reconstructed [65, 66]. The reconstruction is based on searching for polygons with the minimum number of vertices along the predicted orientations. With deep learning based detection method, LOD 1 building footprints are mostly recovered.

The discretisation of an input raster to a binary map during semantic segmentation results in the loss of information originally contained in the image [193]. The resulting

shape estimation from the binary output relies solely on the segmentation accuracy. Consequently, failure of conformance of the object segments boundaries to ground truth adversely affects the resulting building boundary shape. To overcome this limitation presented by semantic segmentation, some deep learning methods that aim to recover building polygon boundaries have been proposed [66, 194, 195, 196]. For example, Castrejon et al. [194] defines the object detection task as a polygon boundary detection problem. The model termed Polygon Recurrent Neural Network predicts shape points along an object boundary and creates a polygon from the points. Similarly, Acuna et al. [195] improves on this idea by using an improved learning algorithm based on reinforcement learning. However, the performance of these methods is based on the availability of training data, in addition to unknown performance with city-scale satellite data. In addition, the training is more complex and the output is simple polygons that do not contain holes. Girard et al. [66] propose an approach that aims to aid in polygonization by outputting shape information in a frame field. A frame field contains directions computed at pixel points that have the property of aligning to object or polygon tangent directions. These directions are then used to refine the output shape of the detected buildings.

Deep learning has, to some degree, the ability to classify or label data giving it context. The application of Scene Learning to the direct 3D reconstruction problem has been limited mainly because of limited training data that cater for different contexts. Although there has been tremendous strides in learning-based detection methods, these methods have been largely limited to single image detection and reconstruction [197, 198, 143] or to improve object detection [54]. For example, Saxena et al. [143], proposed a 3D structure reconstruction method from a single image based on a few assumptions. In this method, an image is first segmented into homogeneous regions, ensuring over-segmentation. Each of the segments, termed superpixels, is assumed to be planar. The assumption made is that each of the super-pixel set belongs to the same planar surface if they exhibit similar features, unless they are separated by an edge. Consequently, neighbouring super pixels are likely to be connected. 3D models are realised by modelling homogeneous image regions obtained from the segmented planar regions. Using a supervised training model, plane parameters are estimated using image features and object-to-object relationships. Han and Zhu [199] present another reconstruction algorithm from single images using a

Bayesian posterior model. First, an image is segmented, and the photometric properties of these segments are extracted. Based on a segmentation and sketching algorithm, a primal sketch of the 2D image in the form of curves and regions is retrieved. Consequently, the image is represented as a set of curves and regions. These segments are represented in a graph, which is further partitioned into sub-graphs. From the graphs, based on prior 3D learning, 3D shapes are inferred, thus reconstructing the model. Occluded regions are estimated during the shape inference stage. However, these methods are suited for and tested mostly on terrestrial images of a single object.

Delage et al. [197], present a reconstruction algorithm from monocular images of indoor scenes based on prior knowledge. Prior knowledge is learnt using a Dynamic Bayesian Network. Based on the assumption of the existence of a floor-wall geometry, the floor boundary is detected, after which a reconstructed scene is generated. The ability to detect boundaries and recreate sharp edges in monocular based 3D reconstruction using machine learning is an important feature that can be combined with stereo reconstruction algorithms.

2.5.3.2 Model Driven Boundary Reconstruction An example of a model driven approach is given by Partovi et al. [187], where building shape reconstruction from DSMs is achieved by finding an optimal fitting roof type from a predefined library of five roof types for a given building object. Firstly, ground points are filtered from a DSM initially derived from SGM using satellite stereo pairs. Subsequently, non man-made features above ground (vegetation, water, shadow, and soil) are classified using advanced a rule based fuzzy spectral classification. These classes are then removed from the DSM leaving man-made building structures in a nDSM. Using the Markov chain Monte Carlo technique with simulated annealing, roof types are fitted to the resulting nDSM. One limitation of this approach is the limited number of predefined roof types. Thus, complex roof structures found in urban areas are not fully represented. Furthermore, partial roof detections as a result of occlusions are not reconstructed. Nex and Remondino [58], presents another complete pipeline from DSM generation to roof shape estimation. Normal estimates are calculated for the DSM point cloud using a $5x5$ filter. A grid based filter is then used in separating terrain and non terrain points from the point cloud. In the grid based filters, a multi-resolution grid is

used in identifying ground points where a large grid is defined to extract terrain heights ensuring large buildings do not form part of the terrain, subsequently, smaller grids of the same area are defined to account for blunders. After filtering the ground, vegetation is removed by using NDVI if Near InfaRed (NIR) band is available, otherwise colour and depth information is used. Using the point normals estimated from the $5x5$ filter, a region growing segmentation method is used to identify planar roof facets, whose boundaries are then extracted. Boundary smoothing is then performed using lines and curves that best approximate the planar roof shapes. The planar roof shapes are then merged to approximate the building boundary shapes.

Arefi and Reinartz [89] presents a model driven approach that reconstructs building models from satellite or LiDAR derived digital surface models. Complex building structures are decomposed into smaller subsets based on roof ridges, which simplifies shape approximation. The final model is reconstructed by the merging of the building sub-components to form the original building structure. For LiDAR data, the resolution allows the accurate extraction of ridge lines. However, because of the lower resolution from high resolution satellite images, ortho images are used in detection ridge lines and then used for the subsequent decomposition of the building. Huang et al. [200] presents another method that uses a library of eleven roof primitives that are fit to roof structures. Complex roofs are reconstructed by allowing the merging of primitives to form more complex roof structures during the roof search, which allows for a greater number of roof primitives to be modelled. Other example methods can be found in Reinartz et al. [64]. Similar approaches have been presented that use deep learning to predict the correct type of roof [176].

In some cases, ancillary data such as existing footprint, roads, and cadastral boundaries are used to aid shape detection and reconstruction. The use of ancillary data in shape recovery often presents its own challenges. The use of multi-sensor data results in registration mismatches in the data which often results in reconstruction errors. Furthermore, multi-sensor fusion increases the cost of data acquisition [41].

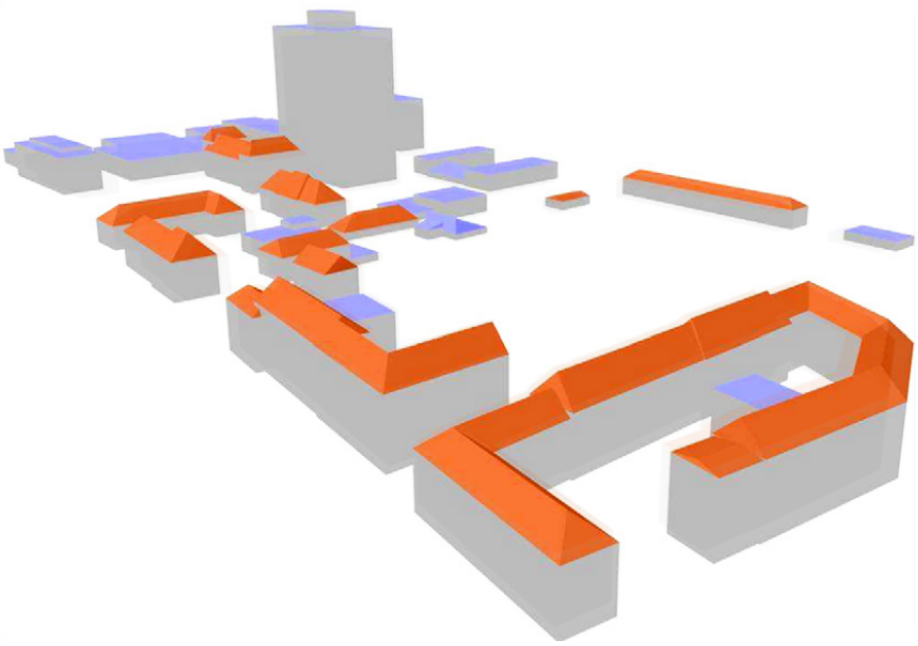


Figure 24: Model driven shape reconstruction from a defined set of building primitives [200]. Roof types are predicted and combined with building footprints and a height to create a 3D model.

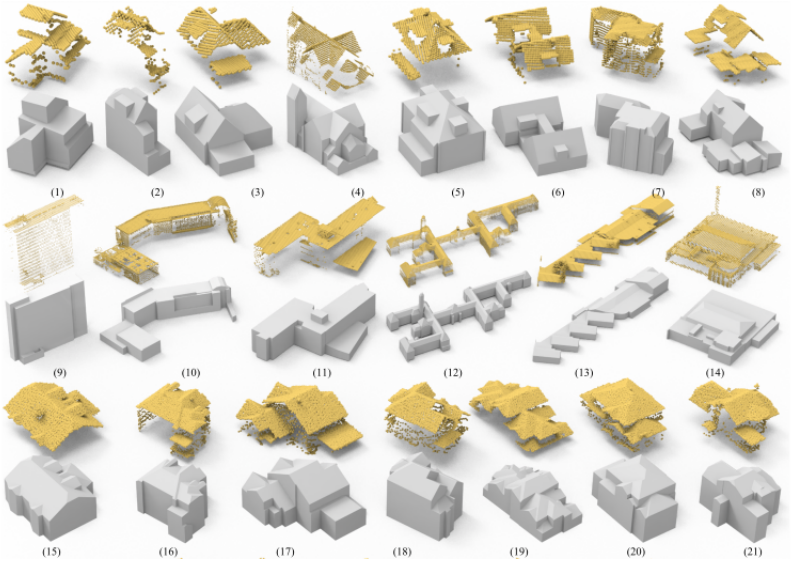


Figure 25: Lidar achieves better building shapes and LOD when compared to satellite imagery models shown in Figure 19 [88]

2.5.4 3D Model Reconstruction

The process of creating the final model is straight forward. Once the shape of the building footprints have been reconstructed, a DSM is used as a height source to create LOD 1 buildings. If the DSM or images have a resolution that allows the mapping of roof structures, then parametric methods are used to recover the correct roof shape and heights

are assigned to the roof structures. This information is used to create the final B-REP model.

A detailed summary of the discussion points raised in this section is given in Section 2.8. Full model reconstruction has remained a challenge when using edge features and lines. Using a novel dense edge recovery framework, 3D reconstruction from detected edges is proposed in this research. The next section summarises the algorithms discussed in this Section.

2.6 Reconstruction Algorithms Summary

This chapter reviewed the most popular algorithms for generating 2D and 3D buildings from satellite imagery. Most algorithms leverage and build on existing algorithms in an effort to provide a complete reconstruction pipeline. Table 1 summarises the abilities of these algorithms in addressing the entire reconstruction pipeline, which is illustrated in Figure 2. The table does not show an exhaustive list, but rather the most common reconstruction methods. Table 1 shows the processing steps followed by these algorithms in detecting and reconstructing building boundary shapes. As illustrated in the table, most algorithms follow the pipeline illustrated in Figure 2. Variations are encountered in the algorithms used for each stage. For example, most of the algorithms shown leverage the semi-global matching algorithm that was first presented by Hirschmuller [57] and modified by various authors [56, 63, 140].

Author	Image Input	DSM generation	Algorithm	Object Detection	Reconstruction	Modular	Comments
Krauss et al. [137]	Yes	No	Dynamic Time Warping	Yes	No	No	Epipolar line segment matching
Zhang and Gruen [48]	No	Yes	MPMM	Yes	No	No	Dense DSM production
Zhang and Fraser [73]	Yes	Yes	Coarse-to-fine hierarchical matching	Yes	No	No	Multi-Resolution Image matching with image features

Author	Image Input	DSM generation	Algorithm	Object Detection	Reconstruction	Modular	Comments
Hirschmuller [57]	Yes	No	SGM	Yes	No	No	SGM for dense point cloud production
Jiang et al. [186]	No	No	No	Yes	Yes	No	Object Detection and labelling
Nex and Remondino [58]	Yes	No	MicMac	Yes	Yes	Yes	Focus is roof shape Recovery-Uses existing matching algorithms
D'Angelo and Kuschk [160]	Yes	Yes	Modified SGM	Yes	No	No	Modified SGM
Rothermel et al. [139]	Yes	No	Modified-SGM (tSGM)	Yes	No	No	Dense DSM Production
Grigillo and Kanjir [100]	No	No	No	Yes	Yes	Yes	Scene classification (3 classes)
Demir and Baltsavias [60]	No	No	No	Yes	Yes	Yes	Scene classification (3 classes) and Data driven roof reconstruction
Qin [24]	Yes	Yes	Modified SGM	Yes	No	No	Modified SGM [139]
Partovi et al. [61]	No	No	No	Yes	Yes	Yes	Requires existing DSM
Reinartz et al. [64]	Yes	No	SGM and TGV	Yes	Yes	Yes	Change Detection in 3D is presented and Model driven roof shape detection
Bethmann and Luhmann [140]	No	Yes	SGM in object space	Yes	No	No	SGM in 3D space
Facciolo et al. [50]	Yes	No	Modified-SGM	Yes	No	No	Multi-Date Stereo Matching

Author	Image Input	DSM generation	Algorithm	Object Detection	Reconstruction	Modular	Comments
Widyaningrum et al. [59]	No	No	No	Yes	Yes	Yes	Data Driven roof shape extraction
Buyukdemircioglu et al. [97]	No	No	Deep Learning	Yes	Yes	No	Building Footprints
Chawda et al. [171]	No	No	Deep Learning	Yes	Yes	No	Building Footprints
Bauchet et al. [65]	No	No	Deep Learning	No	Yes	No	Building Footprints
Girard et al. [66]	No	No	Deep Learning	No	Yes	No	Building Footprints

Table 1: Reconstruction Algorithms Summary and the processes that are addressed by each. Modular refers to the ability of each algorithm to address the reconstruction steps independently.

2.7 Review of Edge and Line Detection

A review of the current building reconstruction has been given. Most of the strategies use edges or lines in detecting and reconstructing buildings. Edges have been utilised as support features in most of the methods. The aim of this research is to achieve 3D reconstruction using edges as the main feature. Consequently, a review of edge and line detection is required. In addition to the gaps identified with building reconstruction, this review identifies the commonly used edge and line detectors, and current challenges. The review highlights the areas of focus for this research and the proposed solutions. First, a review of edge detection is given, followed by line detection.

2.7.1 Edge Detection In Satellite images

Sharp changes in intensity occur because of changes in the surface properties of adjacent structures, or features. Edge detection is the process of identifying areas of sharp intensity changes. Sharp intensity variations are generally attributed to the existence of an object's edge [201]. Since object boundaries are often associated with high-intensity variation,

edges are used to detect the existence of objects. In this section, a review of basic edge detectors that are used in this research is given. More detailed reviews on edge detectors are found in [202, 70, 75]

The edge detection process often begins with image smoothing. This reduces the effects of noise on the detection process by suppressing false large intensity variations. The most popular smoothing operators used are Gaussian smoothing and the bilateral filter [203]. Smoothing aims to reduce high-frequency noise emanating from high signal-to-noise ratio. The level of smoothing has a direct impact on the performance of any detector. Image smoothing may cause information loss, loose edge blending, contrast reduction, edge smoothing, and edge displacement [70]. To minimise any of the aforementioned effects on edge detection, optimal smoothing parameters need to be selected. Approaches such as the variable filter presented by Ali and Clausi [204] are examples of attempts to reduce the effects of smoothing while improving edge detection.

Following the smoothing operation, a gradient operator is applied to retrieve gradient magnitude and direction. Gradient magnitude and direction are obtained by computing a first or second-order derivative on a greyscale image [201, 6]. Finally, edge pixels are identified by thresholding the magnitude of the gradient. Due to factors such as scene variability, imaging conditions, and sensor performance, threshold values vary between scenes. The lack of global thresholding values presents a challenge for large scenes with varying contrast, such as satellite images.

The most popular gradient-based detectors are the Sobel, Roberts, Prewitt, and Scharr operators [183, 205, 206, 207]. Alternatively, second order derivative based detectors such as Laplacian zero crossing, Laplacian of Gaussian and Hough transform exist [201]. However, the main difference lies in the mathematical complexity of the higher-order derivatives compared to the first-order derivatives [208, 209]. The edge direction is derived from the directions computed from the gradient operators. Detected edge pixels are often detected as thick object boundaries. An edge thinning operation is required to reduce the detected edges to a pixel-wide edge width. Good localisation must be maintained during the thinning step. The thinned edges are finally chained to create boundaries [1, 69]. Most edge detectors are built on the aforementioned gradient operators [182, 69]. Statistical thresholding methods, or adaptive thresholding, are

applied to gradient operator results to obtain binary edge maps [178, 210]. Some methods derive a vector map directly from the gradient map. An example of such a vectorisation method is presented by Topal and Akinlar [69]. A vectorised map or a pixel-wide binary edge map is created directly from a gradient image.The method uses the gradient magnitude to select anchor pixels from the entire image. The anchors are then joined using gradient orientations computed from the gradient image using a smart routing technique. One of the most desirable properties of this edge detection method is the ability to produce a vectorised map of detected edges.

Although the use of gradient magnitude to detect edges is the most popular approach, other approaches exist. More recently, learning-based edge detectors have been explored as an alternative [211, 212, 213, 214, 215, 216, 202]. Although similar performance can be derived from deep learning edge detectors, the main limitation is the provision of an extensive training set that represents all edge cases. Other proposed edge detectors can be found in Adamos and Faig [185], El-Sayed et al. [217], Smith and Brady [218]. It is worth mentioning the canny edge detector, as it is the most widely used edge detector [182]. The canny edge detector outperforms most detectors because of the ability to detect edges with reduced fragmentation by using two gradient thresholds. The upper threshold ensures that salient edges are detected, while the lower threshold ensures that weak edges are not missed. Additionally, a non-maxima suppression step ensures that pixel-wide edges are detected, thereby avoiding thick edges.

Image smoothing also requires defining an optimal smoothing parameter in addition to gradient thresholding. A single set parameter does not suffice for large-scale satellite images with varying illuminating conditions and contrast, shadows, and homogeneous surfaces. Some methods counter these effects by using adaptive or automatic thresholding techniques [1, 2]. In Lu et al. [1], the canny edge detector is used with an adaptive thresholding technique to define the upper and lower thresholds required by the canny edge detector. The maximum meaningless gradient and the minimum meaningful gradient thresholds are calculated based on the input image according to the Helmholtz principle [156]. Akinlar and Topal [2] avoid setting parameters by running the original ED algorithm [69], with gradient magnitude parameters set at their extrema. The results of these two automated parameter tuning algorithms are evaluated in Chapter 7. Figure 26 shows results from a satellite image using CannyPF

and (b) EDPF. Missing edges and fragmented edges are evident in the low contrast and dark regions.

Figure 26: Results from (a) CannyPF in red, and (b) EDPF in orange, on a satellite image. Edges in the low contrast and dark colored regions are not detected and edges are fragmented.

The task of selecting the optimal parameters is generally left to the user in most cases. Deducing the correct parameters to use to get optimal results is not straightforward. Consequently, edge detectors often miss good edges that fall outside of parameter ranges. The novel edge detection framework presented here aims to solve for this shortcoming. By running edge detection on a wide parameter space using an accumulator, edges are aggregated thereby, detecting longer unbroken edges. The accumulator framework takes the onus of selecting optimal parameters off the user. Using the accumulator, better edges are detected in large images with varying illumination, contrast, and shadows.

2.7.2 Line Detection In Satellite images

The use of edges and lines during reconstruction [96, 100, 92] has been discussed in section 2.5.3. The presence of line features is associated with the presence of building objects. Line features provide geometric and structural information about objects in a scene compared to point features [82]. Therefore, lines are preferred for object detection and boundary reconstruction. Various works on line detection have been presented and reviewed [82]. The main goal of most line detectors is to preserve both the length and orientation of lines. This section will discuss popular state-of-the-art line detectors with the aim of identifying gaps and areas of improvement.

Binary edge maps or gradient images are often used as input in line detection methods [5, 4, 6, 3]. The challenges associated with edge detection are often transferred to line detection. Fragmented lines, missing lines, and incomplete lines often characterise the line detection results. Furthermore, uncertainty in determining the exact end points of the line affects downstream processes such as segment description and matching and boundary delineation [82]. Improving line detection should therefore reduce some of the aforementioned challenges. Figure 27 highlights some of the challenges with line detection from satellite images.

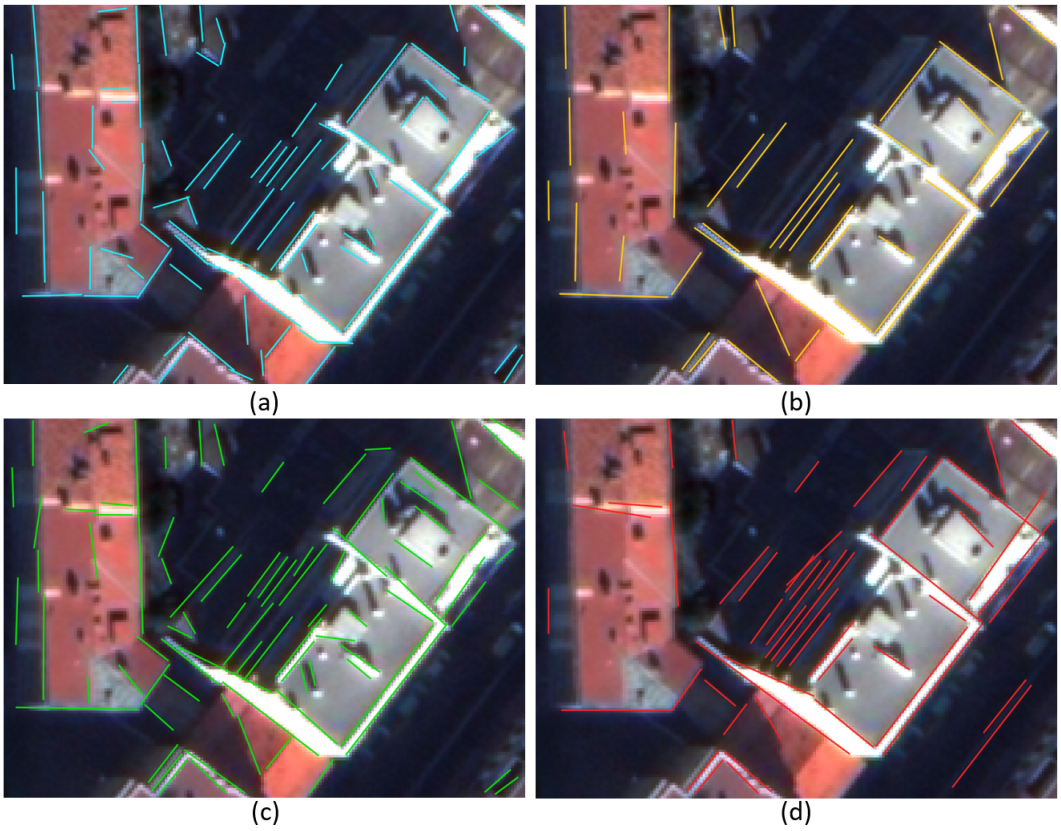


Figure 27: Results of various edge line detectors on a satellite image. (a) are results from LSD [4], (b) are results from CannyLines [1], (c) are results from EDLines [3] and (d) from AG3Lines [6]. The detectors were run with their default parameters. Different levels of fragmentation is evident in each result with lines in the dark regions missing.

Early line detectors were mainly bashed on the Hough transform where intersection points in the Hough space are used to detect lines. Methods such as the Hough transform [219] leverage on a computed edge map to estimate the best fitting line using a voting scheme in the Hough space. Each edge pixel's contribution to a line configuration is assessed during voting and a final line selected. However, in areas made of rich textures, dense edges, or small objects, many false positives are returned. This implies reduced

performance in noisy images. The major challenge with Hough-based methods is the inability to detect line end points with various solutions proposed [5, 220, 221].

Gradient orientation and magnitude based methods provide an alternative of Hough Transform based detection methods. With gradient orientation and magnitude based methods, lines are detected by identifying pixel groupings with similar orientations and magnitude. An example is LSD, a popular detector based on gradient orientation [4]. LSD performs line detection without the need for parameter tuning. To identify candidate lines, LSD clusters pixels based on level line orientations. Pixels that have the same level line angle within a certain error threshold are grouped into clusters named line support regions. A minimum bounding rectangle, centred at the cluster centre of mass, and oriented along the first inertia axis is computed. The identified rectangle along with the pixels associated is used to verify or reject detected lines. The rectangle validation procedure is based on the Helmholtz principle [156] and *a contrario* approach [4]. By computing the number of false alarms associated with a rectangle, valid lines are recovered. LSD attempts to reduce fragmentation in two ways, namely, image re-scaling, where the image is scaled down to 80% of the original size, in addition to a Gaussian filtering kernel that reduces noise. Second, by commencing line search on pixels with highest gradient, the effects of noise on line orientation are reduced, thereby returning better line approximations. Furthermore, rectangle refinements are applied to reduce curve to line approximations, where a curve is approximated by a straight line, resulting in the curved nature of the line being lost. To improve line orientation, borders pixels in a rectangular region are recursively added while recomputing the number of false alarms. LSD, however, usually fails to recover the full length of the line, and noisy pixels at line end points still affect the orientation of shorter segments. There is increased fragmentation in lower resolution imagery such as satellite images, where there are slow gradient changes along object edges. Salaün et al. [222] improves on LSD by using a multi-scale approach that improves line detection while reducing fragmentation. Scaling down an image reduces the resolution. Lower resolution images are used to detect longer lines while higher resolution images are used to improve localisation of the originally recovered longer lines. Longer unbroken lines help to reduce fragmentation.

Cho et al. [223] proposes a Linelet-based line detection algorithm which uses the

intrinsic properties of linear segments projected onto a raster. The assumption is that a projection of a line onto an image space results in a set of pixels that are connected horizontally or vertically. The size or length of the pixel chunks depends on the angle of the line. These chunks, termed linelets, are connected by a periodic stair-stepping. Linelet properties that are unique but not independent of gradient magnitude and direction are computed for each linelet. By grouping linelets based on the computed properties, lines, together with their associated orientation, are detected. The major advantage of this method is that the geometric descriptors of linelets are constant regardless of the variations in image contrast. The intrinsic properties of the lines can thus be recovered even in low-contrast regions. However, for low-resolution images such as satellite images, where edge localisation is not accurate and pixel missing exists, the inter-linelet relationship is easily violated.

Another method that performs well termed Active grouping and geometry-gradient combined validation Line Detector (AG3Lines) is presented by Zhang et al. [6]. In this method, pixels are grouped based on line geometry and alignment of gradient orientation. Instead of starting with pixel clusters to estimate a line, anchor points are selected as line starting points. Anchor points are assumed to exist on a line segment. The anchors are then grouped according to predefined properties. An initial line segment estimated using gradient-orientation, is used and updated actively by incrementally adding anchors. A desirable property of AG3Line is the constraint that anchors must not be contiguous. This greatly reduces line fragmentation due to noise or feature overlaps, such as overhanging trees.

Another line recovery approach presented by Lu et al. [1], builds on an improved canny edge detector [182], termed Parameter Free Canny Edge Detection (CannyPF). The original Canny edge detector improves detected edges by providing two thresholds for selecting edge pixels. However, the lack of global thresholds becomes a limiting factor in selecting the most ideal parameters. To counter this, Parameter Free Canny Edge Detection (CannyPF) was developed, which automatically sets the required canny thresholds based on image quality [1]. The thresholds are determined based on the Helmholtz principle [156]. Similar to LSD, edge pixels recovered using CannyPF are clustered according to gradient orientation. A split operation is applied to a cluster where the maximum deviation from a line formed by connecting the cluster end points

exceeds one pixel. The output segments from the splitting are then merged and extended using a least squares line fitting approach. By merging with near-collinear lines or pixels using the initial line segments from the fitting, the lines are merged and extended. During merging, an error threshold of one pixel is maintained. The lines are finally verified using the Helmholtz principle. Even with automated thresholding using CannyPF, meaningful edges are lost with edge pixels that do not fall within the computed Canny thresholds. Consequently, fragmentation and missing lines are still evident.

Edge Drawing Lines (EDLines) [3] is another gradient orientation and magnitude that takes advantage of the output of the Edge Drawing (ED) algorithm [69]. The ability of Edge Drawing (ED) to extract a pixel-wide edge map benefits the line recovery step. By starting with anchors, which are pixels of high gradient magnitude, and connecting them using a smart routing algorithm through other edge pixels, a clean skeleton is produced. This eliminates thresholding and skeletonisation processes that are usually applied to other gradient-based methods to recover single pixel width edges [182, 183, 224]. By walking the created edge chain and fitting lines, linear segments are recovered. Lines are extended incrementally along the chain by adding contiguous pixels in the chain while tracking the fitting error threshold. However, since the initial edge map is based on thresholding gradient magnitude, edge pixels in low-contrast images are lost, and this results in fragmented and missing edges. ED is sensitive to noise, which causes gradient magnitude jumps, and since the edge direction is based on the gradient magnitude, line orientations are easily affected [1]. Figure 28 shows four overlaid detector results. Each detector is characterised by fragmentation and missing edges.

Line detectors based on deep learning are currently being proposed for line detection [225, 226, 8]. Learning-based techniques rely on sufficient training data. The main limitations of learning-based detectors are the lack of sufficient training data on a large scale and the need for retraining with changes in image resolution. Publicly available training and benchmarking datasets exist for deep learning wireframe parsing by line detection [225, 223]. However, the data set contains mostly high-resolution, small-scale images of individual building objects. The suitability of learning-based methods on high resolution satellite imagery requires investigation.

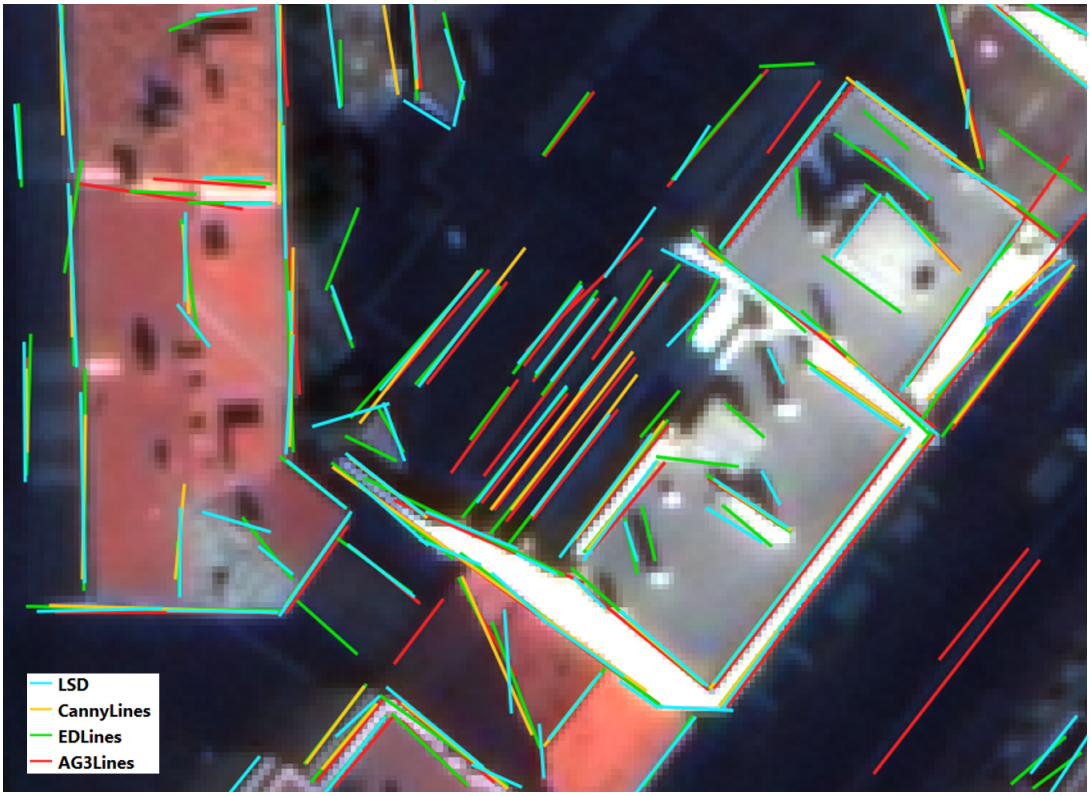


Figure 28: Overlay of four results from LSD, CannyLines, EDLines and AG3Lines. Each method performs differently and some line orientation errors are evident. The detectors all perform differently with some lines missing that are detected by other methods.

2.8 Summary

From the review presented, it is clear that considerable research has been ongoing in the field of building object detection and boundary shape recovery. Various methods take advantage of different geometric and radiometric attributes contained in the input data, such as edges, linear features, radiometric differences, and height information in the case of multiple image views. One of the main goals of reconstruction is to preserve linear and curved edges to reduce shape distortion in the final reconstruction. Most methods aim to recover objects with high fidelity boundaries. However, true edge recovery to aid in boundary detection and, subsequently, boundary shape reconstruction still has some challenges. Recovery of unfragmented edges from an image remains a challenge. The improvements and variations in the different methods are mainly aimed at providing alternative methods that reduce the effects of fragmentation and missing edges.

Fragmented and missing edges still prohibit the use of edge features solely for shape reconstruction and shape recovery [227]. In addition to fragmentation, the completeness

of boundaries is often interrupted by inter-object structures and occlusions [80, 82]. This adds to the complexity of using edges in reconstruction. Rather, edges are used mainly as support features to reduce matching ambiguities and preserve breaklines [73]. In an effort to counter issues arising from missing edges, other non-edge detection methods, such as semantic segmentation and image classification, are utilised to further enhance boundary recovery [51, 188]. The lack of a generic edge detector is another challenge because different edge detectors require different parameter thresholds that vary between images. Consequently, full reconstruction based on edge features is less preferred. This is evidenced by most methods that prefer multiview imagery to derive DSMs, which are then used as the basis for shape extraction [51, 187, 64]. However, DSMs from satellite images are sometimes produced at a lower resolution compared to the input image resolution. This results in boundaries of reduced fidelity. Furthermore, sharp object edges are not completely preserved with DSMs.

In addition to limited edge recovery, the main edges used in boundary estimation are linear [4, 3, 6, 219]. This introduces a limitation when recovering complex shapes. This is because methods that use linear edge features operate under the assumption that objects in the scene are mainly composed of linear segments. This limits boundary recovery methods to rectilinear structures at the expense of curved structures [60, 100, 58, 64, 59]. Furthermore, detection of parameterised lines often requires a minimisation approach, which at times rejects valid lines. Lines are recovered after a prior edge detection process. Consequently, challenges with edge detection are evident in the line detection process. Put differently, line detection quality and quantity depend on edge detection results.

DSMs are generally preferred because they correct for the projective distortion of object structures in multiview scenes. As previously stated, the major drawback of DSMs is the lack of semantic information about objects and geometric information, in addition to the 3D heights generated from the point cloud or DSM. Post processing on the DSM is required to extract 3D objects and boundary shapes [59, 64, 89]. Sometimes, post-processing of the DSMs does not incorporate the original image information that can improve shape extraction, such as edge attributes. Loss of edge and corner information with DSMs further reduce boundary shape fidelity. Some methods use a fusion approach in which an attempt is made to recover information from the original images and use them with the DSM. More recently, semantic segmentation methods, though reliable, produce

binary maps that represent building extents with missing object detail. Consequently, shape recovery is limited to footprints.

Most building detection reconstruction algorithms work well on LiDAR DSMs when compared to image matching derived DSMs [100]. This can be attributed to the high point cloud resolution achieved by LiDAR because of the ability to better capture discontinuities. The point-cloud resolution and accuracy allow for accurate and efficient detection of regular geometric shapes whose Boolean operations result in the most building structure shapes. In contrast, satellite-derived point clouds do not have the benefits that LiDAR data possess. An example roof boundary reconstruction from a DSM is shown in Figure 29. Most of the recovered shapes require improvements. Point mismatches, height accuracy, and reduced resolution present a challenge when detecting objects from photogrammetry derived DSMs, specifically satellite images. However, most of the object geometric features are contained in the raw imagery that is acquired in the form of edges. For accurate building boundary shape reconstruction, this information can be extracted at image level and used directly for shape recovery.

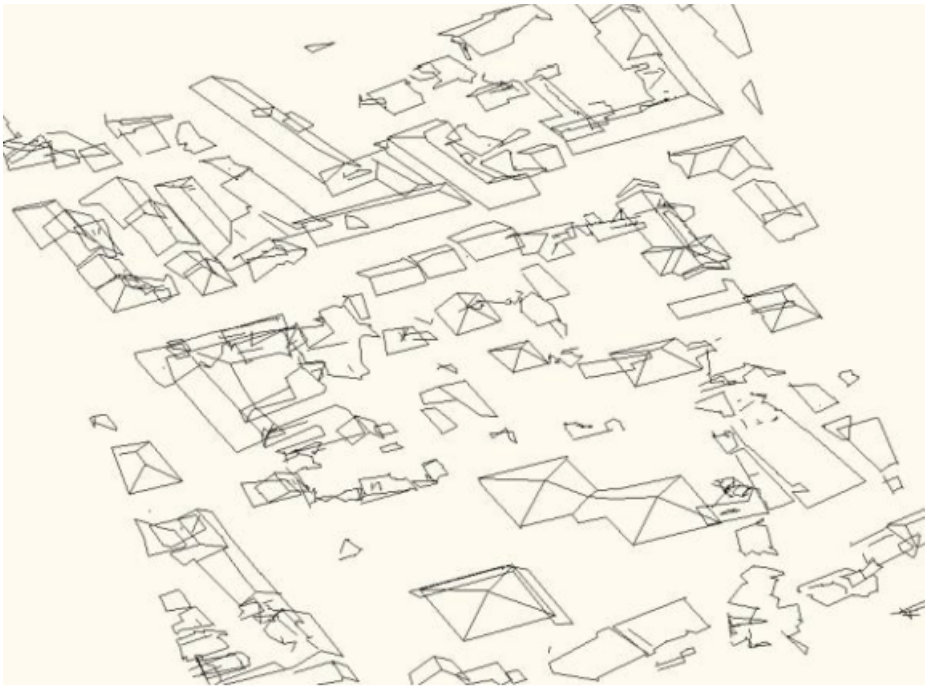


Figure 29: Roof outline extraction from DSMs presented by [58]. While most shapes have been extracted, some of the shapes require improvements.

Traditional edge detectors fail to recover dense edges due to various factors, such as low image contrast, inter-scene radiometric variations, and gradient threshold

requirements. Consequently, traditional edge detectors are only used to provide initial edge features and attributes that aid more robust methods. However, if an efficient edge recovery method is developed, the use of edge features for object boundary shape recovery is possible. The research proposed here aims to reconstruct building boundary shapes with edges as the main feature, reducing reliance on point processes such as pixel matching. By developing an efficient edge detection framework that recovers maximal object boundaries, object boundaries can be recovered and individually matched to produce 3D models of building structures. This will avoid the need for filtering steps to obtain nDSMs. Maximal boundaries are recovered using an accumulator-based edge detection framework. The edges are used to recover lines and curves with reduced fragmentation. The lines are then used in the final boundary detection and reconstruction step.

2.9 Conclusion

This chapter introduced the main concepts of building boundary reconstruction from satellite images. The chapter gave an overall description of the main processing steps required in building boundary shape reconstruction from images. The purpose of the chapter was to provide an appreciation of the various concepts and steps that are followed in building object detection and reconstruction, and highlight the focus areas of this research. The chapter also presents a review of current building detection and boundary shape reconstruction strategies, building on the concepts presented in this chapter. From the review, areas of improvement and gaps are identified, and the new frameworks are formulated. From this review, existing gaps were identified and summarised in Section 2.8. The next chapter gives a summary of the method proposed in this research to solve for some of the gaps identified.

3 Method

This chapter introduces the method followed in developing the edge-based boundary shape reconstruction framework. This framework is composed of two sub-frameworks, namely, the edge detection framework (ABED) and the building boundary shape reconstruction framework (ABEDL). Edges are used as support features in many reconstruction methods. Challenges such as fragmentation and missing edges, have limited the exclusive use of edges for reconstruction. The method developed here aims to solve some of the challenges associated with edge detection and boundary shape reconstruction using a novel accumulator-based edge detection framework and building boundary shape reconstruction framework. The steps followed in the method, namely edge detection, boundary detection, boundary reconstruction, and building boundary shape reconstruction, are discussed. Figure 1 shows the detailed workflow followed. As a reminder, Figure 30 shows a summary of the steps followed in the method. The method developed is classified into three steps, namely dense edge detection described in Section 3.2, boundary detection and reconstruction, which is discussed in Section 3.3, and building boundary shape reconstruction detailed in Section 3.4. Dense edge detection is achieved through ABED. ABEDL implements solutions for boundary detection and reconstruction, and building boundary shape reconstruction.

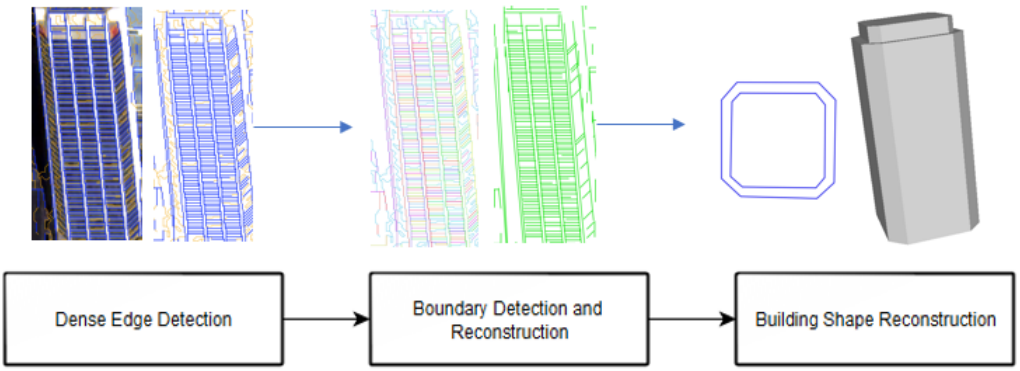


Figure 30: Developed method for building boundary shape detection and reconstruction. The method begins with dense edge detection followed by boundary detection and reconstruction. Finally Building shape reconstruction is done as the final step.

After data sourcing and preparation, the method begins with the development of a novel accumulator-based edge detection framework. The accumulator-based edge

detection framework (ABED) creates a dense edge map as output that is used as input in the subsequent boundary detection and reconstruction step, which is part of the building boundary shape reconstruction framework (ABEDL). During boundary detection, lines and curves that constitute building boundaries are detected using lineset grouping and then classified based on orientation. The recovered lines are connected to form long curves or longer boundaries that follow object form and detail using a BFS algorithm with parameter relaxation. The connected lines are used to recover closed boundaries using planar face traversal. Furthermore, line matching is done to recover the final 3D model wireframes. Finally, quantitative and qualitative results are presented and analysed, followed by a discussion of results, conclusions and recommendations. Each of the steps illustrated in Figure 1 is elaborated on in the following sections.

3.1 Data Sources

Before detailing the method, a description of the data used in this research is required. High-resolution satellite image data with a resolution of $0.5m$ and $0.3m$ is used to develop and test the overarching framework. The data used are images of different areas of different continents, namely Australia, North America, Europe, and Asia. The areas were selected based on the different architectural styles of the buildings. The data contains multiview images acquired at different times and differing off-nadir angles. The image data uses the RPC sensor model and has been corrected for positional biases. A detailed description of the data extents and locations is given in Section 7.1.

3.2 Dense Edge Detection

Edge detection is the process of identifying pixels where sharp intensity changes occur. Sharp intensity changes are usually attributed to the existence of object boundaries. The first step is to detect object boundaries using edge detection. This step consists of four subprocesses that are required to extract a dense edge map from satellite images. The workflow for dense edge detection is shown in Figure 31. The first step is to parameterize the smoothing and gradient functions. Instead of using fixed parameters for smoothing

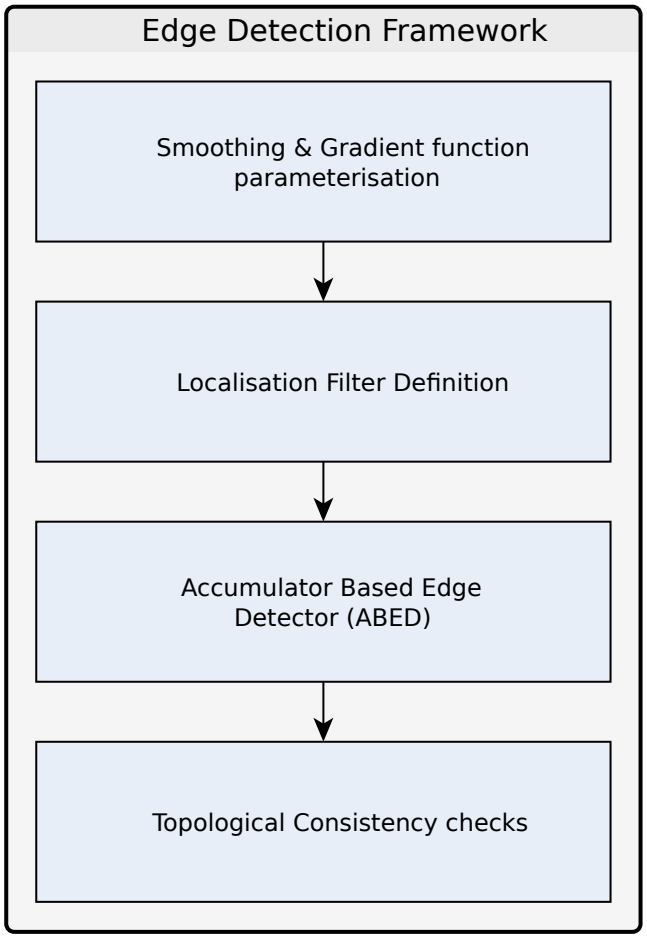


Figure 31: Developed method for dense edge detection. Four sub-processes are implemented in the edge detection framework (ABED) as shown in the workflow.

and gradient thresholding, we redefine the parameters as function variables. This allows the parameter range for both functions to be varied. The second step defines a localisation filter that performs two functions. The first function is to maintain good localisation when aggregating edge maps. The second function is to validate edges before they are accepted into an existing edge map. The third step uses the functions defined in the first and second steps to aggregate and grow edges in an accumulator space that automatically varies the parameters. This edge aggregation step is alternatively termed edge growing. The accumulator space traverses a wide parameter range invoking the redefined gradient and smoothing functions producing multiple edge maps. The edges produced are aggregated and validated to obtain a dense edge map. The dense edge map produced is used in the next step, which is boundary detection and reconstruction. An example dense edge map is shown in Figure 32. The dense edge map is used to create a dense edge graph

that is used as input in the next process. To avoid the dropping of short edge segments, topological consistency is enforced on the graph using direction-constrained snapping. This snapping reduces dangling edges on the edge map. The strength and novelty of the framework is that it is gradient operator-agnostic and aggregated edges regardless of image source, while automatically tuning parameters.

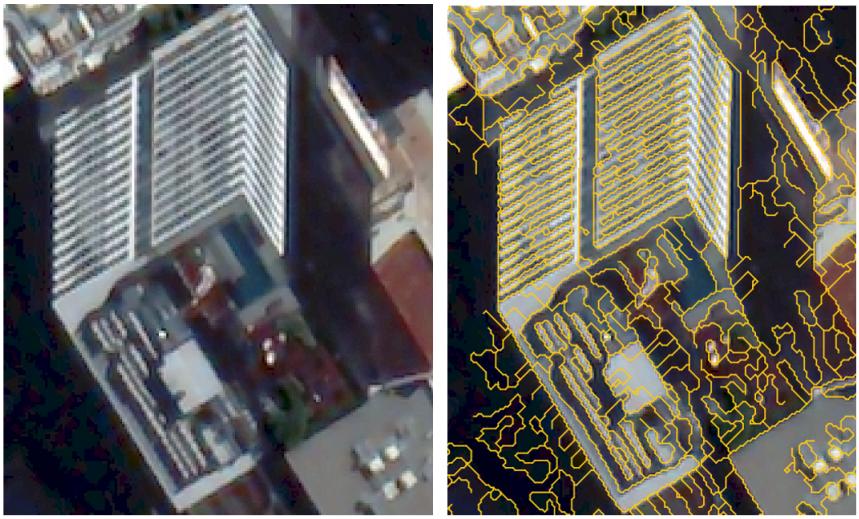


Figure 32: Example Dense edge detection on a high rise building showing both object form (outer boundaries) and detail (windows). Orange lines on the right image represents the detected edges produced from the accumulator.

3.3 Boundary Detection and Reconstruction

The boundary recovery step follows edge detection. The dense edge map produced from the previous step contains edges that represent object form and detail. The edge map contains all salient and weak edges. From this edge map, object boundaries are detected and recovered. This is achieved by detecting and growing lines to form unbroken lines through graph segmentation using line orientation. Segmented lines are further merged in a linear and curved boundary detection step. In this step, the segmented lines are merged into linesets based on orientation and classified. The merging is based on the projective properties of lines on a grid. Linesets are then classified into four classes based on orientation, as detailed in Section 5.3.4. Lines are then fit to the classified linesets using least squares to reconstruct connected lines and curves that constitute boundaries. From the detected lines potential outliers are identified. The reconstructed linear and

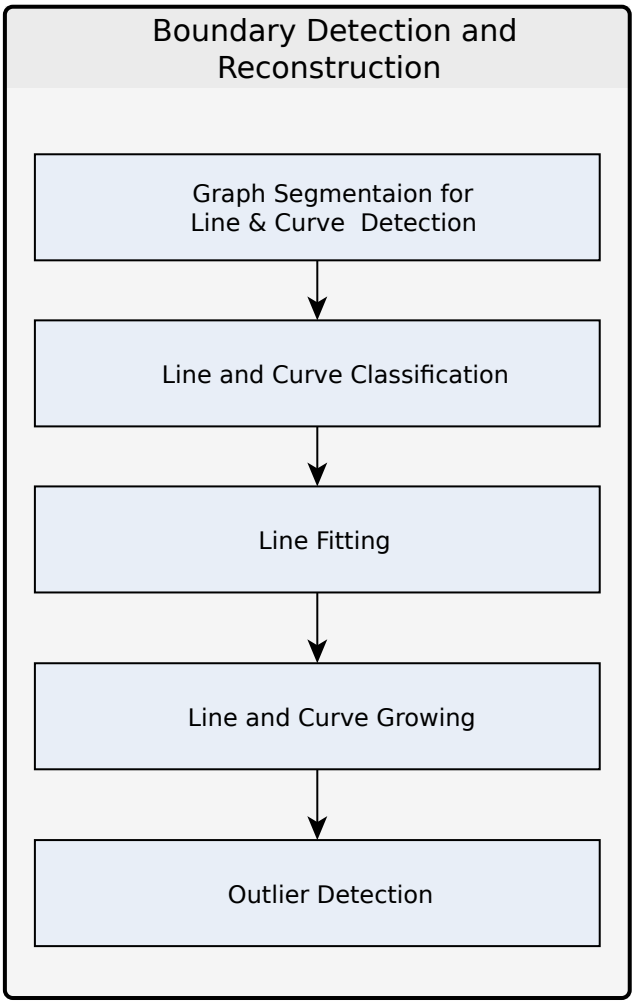


Figure 33: Developed method for Boundary Detection and Reconstruction which is made up of five steps. The input for this method is the dense edge map produced from the edge detection framework presented in the previous section.

curved boundaries are used as input to the boundary shape reconstruction step. An example of boundary detection and reconstruction is shown in Figure 34.

3.4 Building Boundary Shape Reconstruction

Generally, shape estimation of any detected buildings is a prerequisite to 3D building model generation. The lines and curves produced from the previous boundary detection and reconstruction step are connected to retrieve closed or longer object boundaries. This is achieved through a BFS algorithm with parameter relaxation. Starting with a random longest boundary, the BFS function selects a path to traverse based on a minimum incidence angle. During the selection of paths, linear boundaries are aggregated

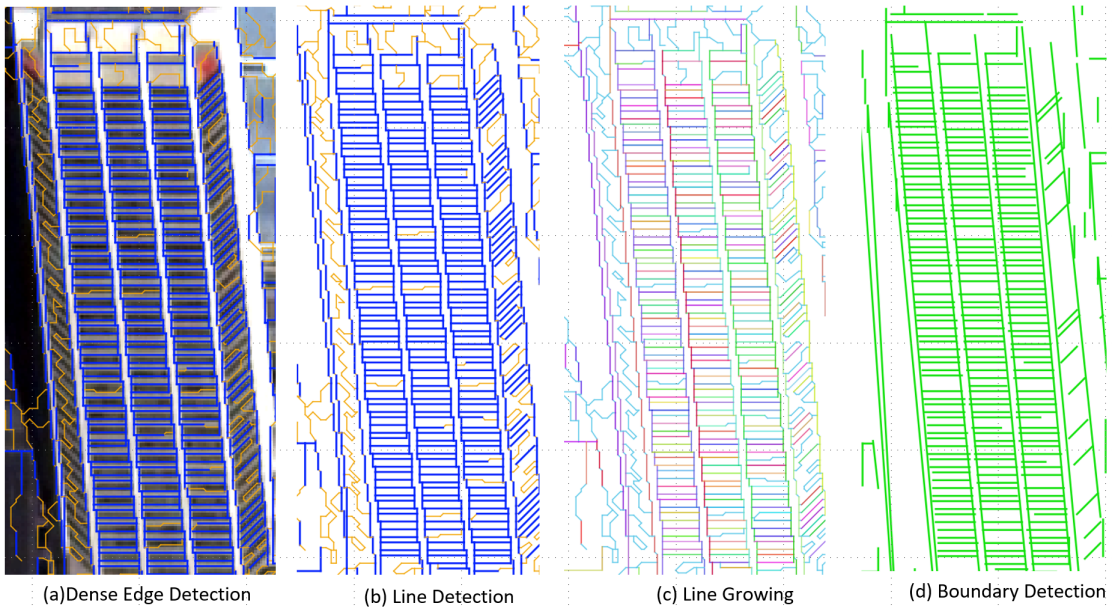


Figure 34: Developed method for Boundary Recovery. (a) shows the dense edge map overlaid with input image. (b) are line segmentation results from classification. (c) is line merging based on orientation and (d) are the final linear segments recovered from line fitting.

based on a linear fitting threshold. The fitting threshold is relaxed in a second iteration, where potential adjacent lines are searched for using a depth parameter, where a single depth refers to a single small edge of less than four pixels. A depth of four implies that four short edges can be traversed from a valid edge with the parameters relaxed until the next valid edge is reached. This relaxation ensures that noisy edges, corners, and small object detail do not result in fragmented boundaries. If a valid line is reached, the two valid edges are considered connected. The novelty of this BFS algorithm is that boundaries are considered as a network of globally connected lines. Consequently, global information is considered during boundary merging and refinement. As opposed to fitting lines independently, as with traditional approaches, connected adjacent lines in the network introduce constraints in the fitting that assist in maintaining correct orientations of lines globally. For example, if a junction point is contained in a set of points, belonging to a particular line, all other lines connected to the junction will maintain their orientation regardless of the fitting result. The merged lines are then used in a planar face traversal algorithm to recover closed boundaries in 2D. In 3D, lines are matched to produce 3D building wireframes.

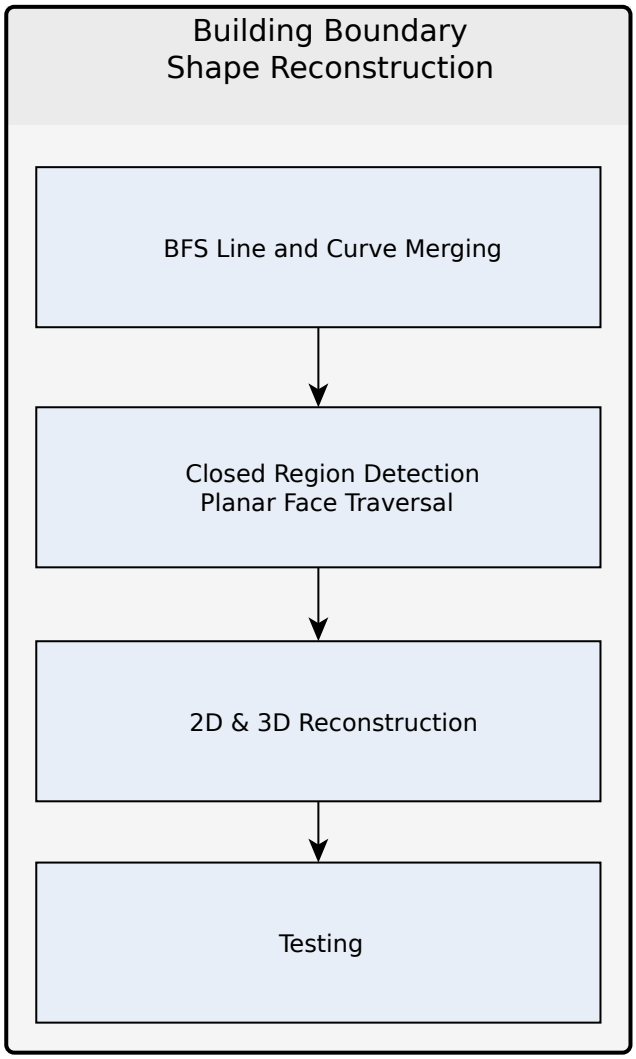


Figure 35: Method for Building Boundary Shape Recovery and Reconstruction from detected lines. The method is composed of four steps shown in the diagram.

3.5 Evaluation Method

Finally, results are assessed both qualitatively and quantitatively. Ten sites containing building structures are used to assess the results. First, for edge detection, the ability of the detector to retrieve both object form and detail is assessed. This is achieved by comparing with the results two state-of-the-art parameter-free edge detectors [1, 2] and ground truth. Second, the lines and curves returned in the boundary reconstruction step are compared with eight state-of-the-art line detectors, five conventional methods [3, 4, 1, 5, 6] and three deep learning methods [7, 8, 9] and ground truth. A quantitative assessment of the length of the line and the completeness of the boundary is done. 3D

wireframes of buildings are generated using the recovered boundaries. A discussion of results is given followed by conclusions and recommendations.

3.5.1 Qualitative Analysis

Qualitative analysis gives a visual assessment of the fidelity of the retrieved boundaries. First, a visual comparison of the recovered dense edges and the buildings is given. From this comparison, the amount of detail recovered can be assessed. Second, a visual comparison of the reconstructed boundaries is presented with an overall assessment of how each of the methods follows object form and detail. The recovered lines and curves are compared with the ground truth to determine the accuracy of the detection. An analysis of failure cases is given and explained in the discussion of results in Chapter 8.

3.5.2 Quantitative Analysis

For quantitative analysis, recall, precision, and F1-Score are used. For edge detection, the recall is computed to assess the amount of edges recovered. Precision is used in this assessment but does not reflect detector perfomance, since ground truth contains only building boundaries, while the edge map contains all edges. The rate of edge growth achieved using ABED is given where the percentage of growth contributed by each accumulator level is given. A measure of line coverage of the retrieved edge, which is a function of length and orientation against ground-truth edges, is given. Ground-truth edges are matched and paired with the retrieved lines. From the pairings, the percentage length of lines detected by various detectors is compared with ground-truth line segments. This comparison also gives a measure of orientation correctness. Recall, precision, and F1-Score will be used to evaluate the results using manually labelled ground-truth images. Furthermore, line detection rate, which is the measure of coverage of a detected line versus the true line in the ground truth, is used to assess boundary completeness. Traditionally, line detection methods make use of F1-Score as an evaluation metric without assessing how individual line length and orientation correctness are. By using Line coverage which is determined by line length and orientation, the performance and accuracy of the boundary completeness can be assessed.

3.5.3 Performance Analysis

An assessment of detector performance with regards to run-time on different datasets is given at the end of Chapter 7. Runtime and complexity are discussed, followed by a discussion and summary of the results.

3.6 Conclusion

This chapter presented an overview of the method followed in this research. The next chapters present detailed descriptions of the steps in the method, starting with dense edge detection, in Chapter 4.

4 Dense Edge Detection

The use of edge features is ubiquitous in object detection and reconstruction [201, 224, 228, 75]. Edge features provide auxiliary geometrical and contextual information on the presence, orientation, and extent of objects. Edges are mainly used to detect and reconstruct object form and detail [99, 229, 230]. However, the use of edges in 3D reconstruction has been limited to support features in most algorithms. This is largely due to the inability to retrieve sufficient edge density that allows full object reconstruction. Edges are characterised by sharp intensity or contrast variations in images. Conversely, insufficient intensity variation or poor contrast results in edge fragmentation, missing edges while image resolution affects edge localisation. Furthermore, neighbouring objects of different hues but similar intensity will miss the edges in between [231]. Efforts to detect weak edges in images have been presented [210, 72]. Because of this, the use of edges as the main feature for object detection and subsequent 3D reconstruction has been limited. Pixel matching approaches have been preferred [57, 56]. Various edge detectors have been proposed in literature each performing differently under different imaging conditions [201, 224, 228, 75, 69, 182]. The main goal of most detectors is to improve the ability to retrieve good edges from a given image while reducing fragmentation and missing edges [70, 209, 72]. Although having multiple edge detectors is desirable, challenges are faced when selecting the best edge detector that suits a given task. Additionally, most edge detectors operate in a defined parameter range. For example, a threshold parameter for filtering out non-edge pixels while retaining salient edges is required for Sobel-based edge detectors [183, 69]. Selecting the optimal threshold requires the end-user to experiment when determining the optimal parameter.

This chapter presents an edge detection framework termed ABED, described in Section 4.1, that aims to leverage existing gradient-based edge detectors to retrieve long unbroken edges, which are termed maximal boundaries. Furthermore, the framework reduces fragmentation and missing edges while maintaining good edge localisation. The detection framework performs under different imaging conditions with any given gradient-based detector thereby achieving genericity. The framework removes the task of selecting optimal parameters or having a-priori knowledge of the image radiometry off the end user. This is achieved by defining a wide and fixed parameter space in an

accumulator. The accumulator produces different edge maps by automatically varying the parameters. The edge maps are aggregated using a localisation filter, resulting in edge growth. The localisation filter additionally validates edges during aggregation. The result is an edge detection framework that retrieves maximal boundaries even under difficult conditions such as shadows and insufficient contrast areas. The accumulator produces a dense edge map. Using the dense edge map, topological consistency is enforced through a direction-constrained snapping filter. This is discussed in Section 4.1.6 . The direction-constrained snapping filter produces the maximal connected edges or boundaries of the object. Finally, a discussion on the developed framework is given in Section 4.3.

4.1 Edge Detection Framework

The challenges faced in edge detection specifically in large-swath imagery have been explained in Section 2.7.1. Cognisant of the fact that many algorithms exist, there have been limited attempts in implementing a multi-detector framework that operates on a wide parameter space of a given detector. Using multidetectors usually results in duplicate edges with inconsistent localisation. This is illustrated in Figure 36 where three different detectors were used with different thresholding parameters.

Despite the presence of spurious or noisy edges in Figure 36, when a multi-detector fusion is attempted, most of the true edges are detected. To solve erroneous localisation and spurious edges, this research develops a generic edge detection framework. The framework is based on a multi-detector approach that uses an accumulator space and operates on a wide parameter space, beyond parameter extrema, to detect and aggregate edges. A localisation filter is used to maintain good localisation during edge aggregation and to validate edges. To ensure genericity, the edge detector is operator-agnostic, implying that any and many gradient operators can be used. To avoid parameter setting on any of the pre-processing required, namely smoothing, a multi-smoothing function approach similar to the gradient-based multidetector approach is adopted. The edge detection framework is termed Accumulator Based Edge Detector (ABED) framework.

As detailed in section 2.7.1, most edge detectors begin with a smoothing or noise

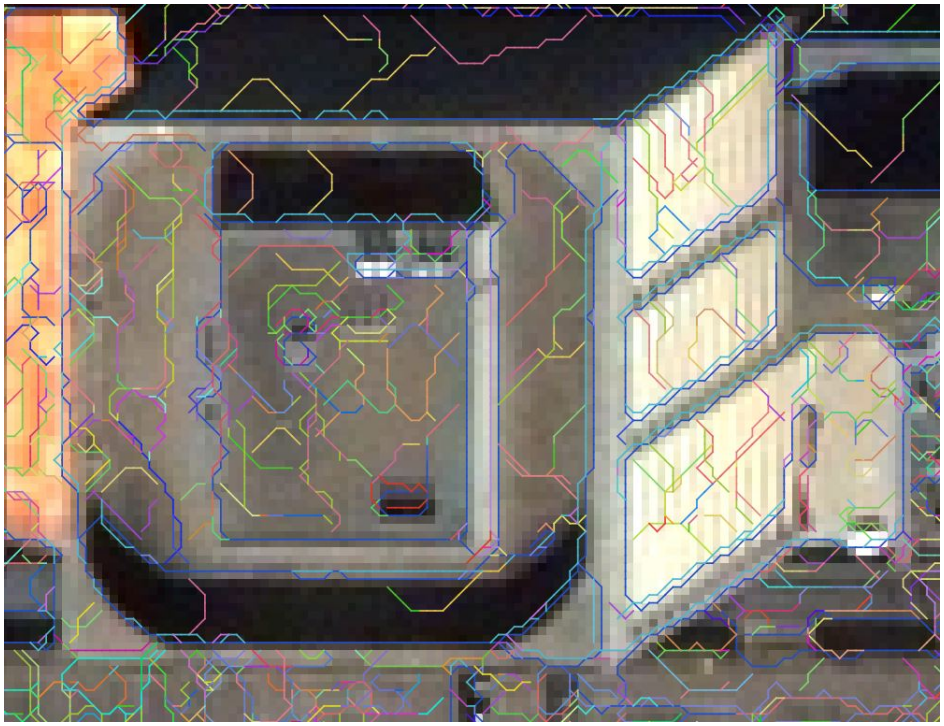


Figure 36: Merging results from three detectors namely Sobel, Prewitt and Scharr with varied parameters affects edge localisation. The different colors represent edges from different detectors using different parameters. Duplicate edges and positional biases are evident.

reduction operation on the image to reduce the effects of noise [1, 69]. The smoothed image is then convolved with a gradient kernel from which gradient magnitude and direction is computed. From gradient magnitude and direction, edges are obtained using thresholding to filter out non-edge pixels. The edge pixels identified are thinned into single-width edge chains that are optionally vectorised. The pipeline of the framework presented follows similar steps. However, to enhance detection, new techniques are adopted at every stage. Firstly, instead of relying on a fixed smoothing function with a pre-defined optimal smoothing parameter, a set of smoothing functions are defined. This set of smoothing functions requires optimal smoothing parameters. Instead of having the smoothing parameter fixed, we redefine the smoothing functions to take the smoothing parameters as function parameters. The same redefinition is applied to the set or family of gradient functions. This redefinition allows the smoothing and gradient functions to receive varied parameters at any given time. After redefining the smoothing and thresholding functions, an accumulator is defined over the smoothing and thresholding parameter space. Iterating over a wide parameter space of these functions allows us to invoke the smoothing and gradient functions with different

parameter combinations. Each parameter begins with the highest value. A parameter combination is invoked at what is termed an accumulator level. Each combination of parameters produces an edge map or edge layer in the accumulator space at each level. The first layer in the accumulator space is termed the anchor layer at level zero, which is considered the highest accumulator level. Successive edge layers that are produced are merged with the anchor layer at each accumulator level. To avoid the errors that accompany edge aggregation, namely edge duplication and incorrect localisation, a localisation filter is defined, which ensures positional consistency of edges from different layers. Furthermore, the filter removes and avoids creating duplicate edges, and validates edges. The result is maximal boundary retrieval for objects in the image. The edges capture both object form and detail. Most algorithms aim to retrieve the form of an object leaving out detail. The Edge Detection framework pipeline is shown in Figure 31.

4.1.1 Smoothing Function Parameterisation

Any imaging process is not immune to image noise. High signal-to-noise ratio results in high frequency noise in satellite images. This usually introduces spurious or false edges in the edge detection process. To reduce the effects of noise on edge detection, smoothing is often required. Edge detection, post image smoothing, has proven to yield better results [201, 203]. Any image smoothing function must be able to preserve edge information while suppressing high-frequency noise. Figure 37 illustrates the effects of smoothing using a bilateral filter. The salt and pepper effects in image (a) are greatly reduced, while preserving edge features in image (b). Figure 38 in image (a) illustrates a noisy image and the effects on edge detection. False edges are introduced as a result of the salt and pepper noise on the roof of the building.

Various smoothing functions exist, but the most popular functions for image smoothing prior to edge detection are edge-preserving filters. The most commonly used edge preserving filters are the Gaussian filter, the bilateral filter, guided filtering, iterative guided filtering, edge-avoiding wavelet, weighted least squares, geodesic editing and domain transforms [201, 232, 106, 233, 234]. The effects of a smoothing operation on edge detection are given by the illustration in Figure 38. Image (a) has more false

Figure 37: The results of a bilateral filter on satellite image data. The salt and pepper effects in image (a) are smoothed out with all edges preserved in image (b).

edges compared to image (b) due to salt and pepper noise. Edges from image (b) where detected after applying a bilateral filter.

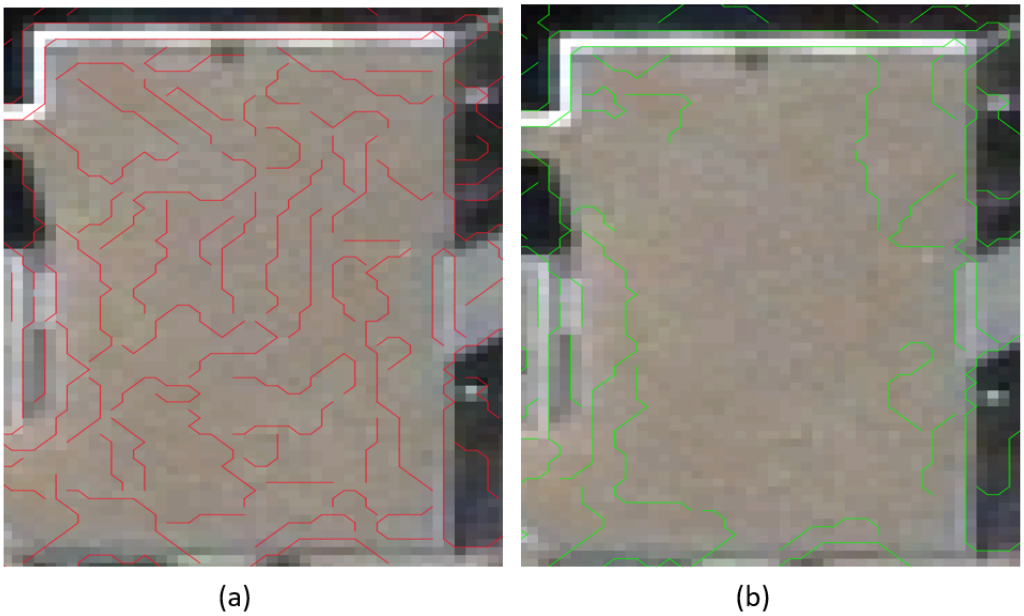


Figure 38: The results of the edges obtained after applying bilateral filter to a building. Spurious edges are detected more in image (a) without smoothing in comparison to image (b), where smoothing was applied before edge detection.

The bilateral filter and its variants [233], have been studied extensively and have now been adopted for various computer vision tasks as a pre-processing step. In this section we present the parameterisation of the bilateral fitler for use in the accumulator space as an example. However, any smoothing function that operates on fixed smoothing parameters can be re-parameterised.

For example, using the bilateral filter, fixed values for σ_s and σ_r are required to define the convolution kernel. The bilateral filter f_{bf} is defined as the weighted average of pixels based on proximity and radiometric values:

$$f_{bf}(I) = \frac{1}{W_p} \sum_{q \in s} G_{\sigma_s}(||I_p - I_q||) G_{\sigma_r}(|I_p - I_q|) I_q \tag{4.1.1}$$

The normalisation factor W_p is computed by:

$$W_p = \sum_{q \in s} G_{\sigma_s}(||I_p - I_q||) G_{\sigma_r}(|I_p - I_q|) \tag{4.1.2}$$

where I = Image channel

σ_s, σ_r = Smoothing parameters

Function G_{σ_s} is a Spatial Gaussian weighting that reduces the influence of pixels based on distance and $G_{\sigma_r}$is a range Gaussian that reduces the influence of pixels with large radiometric difference [233].

The challenge for users is selecting the optimal parameter values σ_s and σ_r that will preserve edge information while suppressing noise. To remove the onus of supplying optimal parameters, we redefine the bilateral smoothing function to operate over a parameter range as opposed to a single value. Therefore, we can define a smoothing function f_{bf} that takes in smoothing parameters σ_s and σ_r by changing the smoothing parameters to function parameter. Using the bilateral filter function as an example, this results in a smoothing function f_{bf}, which is defined as

$$f_{bf} = \{f_\sigma(x, \sigma_s, \sigma_r) \mid \sigma_s \in [a, b]; \sigma_r \in [c, d]\} \tag{4.1.3}$$

where σ_s, σ_r = smoothing parameters as function variables

x = image

This redefinition allows the smoothing function to be applied with different smoothing parameters over a parameter space with range $[a, b]|$ and $[c, d]|$. The range of a ,b,c and$b$

must be large enough to contain the optimal values of σ_s and σ_r, assuming we use the bilateral filter. The same redefinition can be applied to any smoothing function resulting in a set of smoothing functions f_s.

$$f_s = \{f_{bf}(x, \sigma_s, \sigma_r)\} \tag{4.1.4}$$

where σ_s and σ_r = Smoothing Parameters
x = image
s = Number of Smoothing Functions

4.1.2 Gradient Operator Parameterisation

As previously mentioned, most edge detection operators require a threshold that filters out non-edge pixels. Thresholding contributes to fragmentation and missing edges because a global threshold is not applicable to a large image that has noise and varying contrast. To counter this, we redefine the gradient operator, to take in as a parameter the thresholding value. Similar to smoothing, we define the gradient-based operation that establishes the gradient magnitude and direction as:

$$G = g(x, \psi); \psi \in [m, n] \tag{4.1.5}$$

where ψ = threshold value for identifying edge pixels
x = image

This also implies G exists in a parameter range $[m, n]$. The set G_s of redefined gradient functions can now be defined as

$$G_s = \{G_g(x, \psi)\} \tag{4.1.6}$$

where ψ = threshold value for identifying edge pixels
x = image
s = Number of Gradient Functions

Similarly, this implies that any gradient operator in the set can be invoked with different thresholds in a threshold parameter space with range $[m, n]$. The range of m and n must be large enough to contain optimal threshold values of ψ. Different gradient operators can be redefined in the same manner. This allows gradient operators to take thresholding values as function parameters.

4.1.3 Localisation Filter and Vectorisation Function

As mentioned in section 4.1, merging edges obtained from different operators results in noisy and duplicate edges with different localisation. Edges detected by different operators using different parameters rarely overlap. Often, there are positional displacements introduced by smoothing and using different gradient thresholds. In addition, there are missing and fragmented edges as a result of different thresholds being applied. This presents a challenge when trying to grow or aggregate edges. In addition to the positional shift, identical edges are detected in different edge layers. To remove these aforementioned effects, a localisation filter is defined. A localisation filter is a convolution kernel that ensures that the correct location of the edge is maintained during aggregation. In addition, the filter avoids edge duplication and parallel edges that are pixel-width apart. Finally, the filter validates the edges that are added to the edge map.

Directly merging two edge layers results in double and spurious edges. This is illustrated in Figure 39. Edge e_a and e_b is the same edge that has a positional displacement. Aggregating these two edges results in the double edge effect.

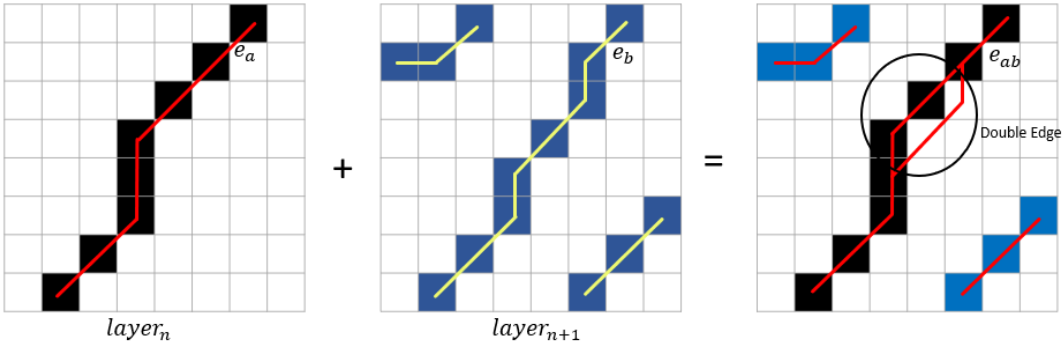


Figure 39: The negative effects of merging edge maps e_a and e_b that have shifted edge positions in their respective layers. The resulting edge e_{ab} has a double edge created which results in noisy edges.

The localisation filter eliminates the occurrence of double and duplicate edges in the merged result. The filter ensures that the desired result is a single continuous edge, as illustrated in Figure 40. The filter works by analysing the neighbouring pixels of a candidate pixel and ascertains if duplicate pixels exist. The noisy and double edge effect that results from merging different edge layers obtained from different operators is eliminated. The filter is also used in the validation function to determine valid edges during edge aggregation. This function is detailed in Section 4.1.4.

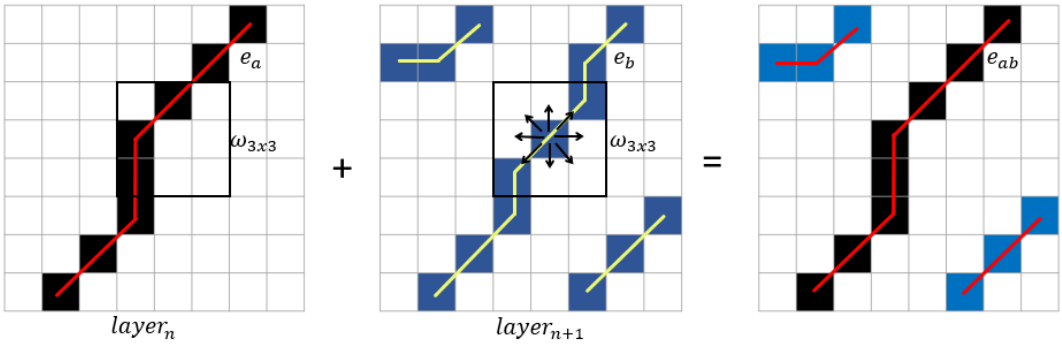


Figure 40: Correct merging is achieved using a localisation filter that maintains correct localisation while avoiding duplicates and parallel edges. The result is a clean edge as shown by resulting edge e_{ab}.

The localisation kernel, ω_{3x3}, used in ensuring edge fidelity during edge growing is defined as:

$$\omega_{3x3} = \begin{cases} 1 & g_x \geq \psi \\ 0 & g_x < \psi \end{cases} \tag{4.1.7}$$

The filter works by establishing the presence of any edge pixels in the filter neighbourhood. The filter returns true if any edge pixel exists in the neighbourhood and avoids the addition of new pixels. The output at any accumulator level is a binary image of edge pixels. The binary image is then vectorised. The localisation filter operates on this vectorised edge map. Figure 41 illustrates an example of a localisation filter applied during edge aggregation. In this investigation, the vectorisation strategy presented by [69] is used. The vectorisation uses a smart routing algorithm based on gradient magnitude and direction. However, any vectorisation method that retrieves single-pixel width edges can be used. The localisation filter ensures that only new pixels are added to the vectorised edge map during aggregation.

The vectorised layer l_{v_t} at level t in the accumulator is defined as.

$$l_{v_t} = \nu_t(g(x, \psi)) \tag{4.1.8}$$

where $\nu_t(g(x, \psi))$ is the vectorisation operation at accumulator level t.

The edges obtained from the vectorisation are aggregated at every accumulator level. The growth function uses an edge validation function explained in Section 4.1.4 to identify new and exiting edge pixels.

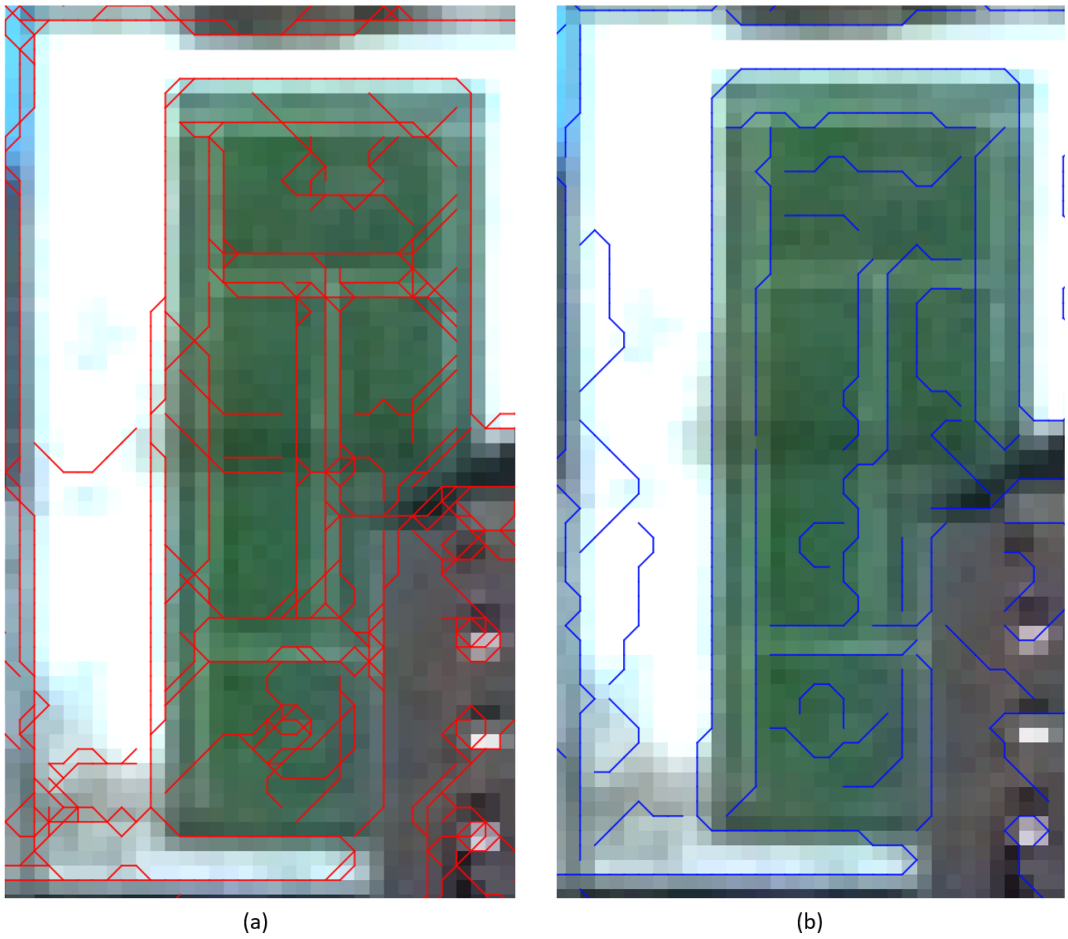


Figure 41: Example results after convolving the image with the localisation kernel ω_{3x3} on edges retrieved from a tennis court. Image (a) shows the double edge effects as a result of merging different edge maps. Image (b) merges using the localisation filter which results in a cleaner edge map.

4.1.4 Edge Validation Function

Parameter variation in the accumulator results in the discovery of weak, spurious, and noisy edges. As a result, only valid edges must be used during aggregation. Edges are validated using the localisation filter, which, in addition to ensuring consistent localisation, is used to ascertain the validity of an edge. The first property that is used to validate edges is that edges rarely occur in isolation. The edges of objects are connected and form closed regions. The second property used is that weak edges are of limited length and are usually in the vicinity of strong edges. The variation of a gradient threshold introduces new pixels in an edge map. By assessing how the new pixels are supported by edges recovered from higher parameter thresholds, edges are validated. Put differently, the anchor layer is used as an initial baseline to grow edges. The anchor layer is continuously grown while being used as a baseline for computing the probability of a new edge being valid. With the exception of the first edge map at accumulator level zero, a new edge always contains edge pixels from the previous accumulator level in addition to new pixels. First, the new set of pixels that have been detected is computed using Equation4.1.9:

$$p_{e_a} = p_{e_n} - p_{\omega_{3x3}} \tag{4.1.9}$$

where p_{e_a} = The new pixels on an edge detected by varying a threshold ψ
p_{e_n} = Valid pixels returned for a threshold ψ
$p_{\omega_{3x3}}$ = Pixels that have been previously detected given by the localisation filter

The accumulator only adds new edge pixels that are anchored by existing pixels that have already been detected from higher accumulator levels. The equation above computes the number of new pixels detected by a given parameter set by removing existing or anchored pixels. The decision on whether to add the new pixels is based on the number of anchored pixels found. This is achieved by using equation 4.1.10. An edge is considered valid if the number of new pixels on the edge is greater than three, and more than half the total number of old pixels are found in the anchor layer. Three is selected because a single pixel cannot define an edge chain and two pixel do not constitute a meaningful edge. Consequently, four pixels are chosen to constitute a valid edge chain. Since four

pixels alone cannot be successfully validated, the four pixels must be the difference p_{e_a} on condition that the percentage of existing pixels found on the edge being validated is more than half the total length of the edge chain. This is shown by the upper term of Equation 4.1.10. The lower term adds an edge if the length of the edge is greater than twice the minimum acceptable edge length of four, under the condition that no other pixels exist in its vicinity. However, edges added by the lower term are flagged as links, whereas those from the first term are flagged as anchor edges and passed on to the next accumulator level.

$$e_{v_n} = \begin{cases} p_{e_a} & \sum p_{e_a} > 3 \mid \frac{\sum p_{\omega_{3x3}}}{\sum p_{e_n}} > 0.5 \\ p_{e_n} & \sum p_{\omega_{3x3}} = 0 \mid \sum pe_{e_n} > 8 \end{cases} \tag{4.1.10}$$

where e_{v_n} = new valid edge

The above-mentioned functions are then implemented in the accumulator space in the edge growing algorithm. This step is detailed in the next section. The major advantage of vectorisation as opposed to finding pixel groupings is that single width edge chains can be processed efficiently during aggregation or merging. In addition, connection priority is given to pixels with higher gradient magnitude during vectorisation. This reduces the inclusion of spurious neighbouring pixels of lower gradient magnitude that are not connected to salient edges.

4.1.5 Accumulator Based Edge Detector

Since many edge detection techniques for edge detection exist, the user is tasked with selecting a suitable edge detector. In addition, optimal parameters for smoothing and thresholding, which are all data dependent, are required. To avoid experimenting with different operators and parameters, an accumulator based approach is developed. The smoothing and gradient functions are redefined to take as parameters the smoothing and thresholding values, as opposed to using constants. The inputs to the accumulator function are smoothing and gradient functions, localisation and edge validation filters. These functions can also be pre-set in the accumulator and not passed as parameters.

Initially, the accumulator space is empty, then starting from the highest values in the parameter space for gradient threshold and lower values for the smoothing functions, an edge map is produced for each pair of parameters. The first edge map, or edge layer, is termed the anchor layer with valid edges, since we know that higher values of ρ and low ψ values yield accurate edges albeit with high edge omissions. The anchor layer is created at the accumulator level zero.

Iterating through the parameter sets creates a layer with every parameter combination. When a new layer is created at every accumulator level, the anchor or previous layer is aggregated with new edges found in the current layer level, while validating the new edges using existing pixels. The validation step can be viewed as a voting process. Voting is an accumulation of the number of hits a new edge chain receives at each layer level. The hits are the number of existing edge pixels from the previous layers. Edges are grown if, for an existing edge, partial votes are received for the new edge leaving candidates in the new edge without votes, which implies a new edge section that has been identified. This voting is calculated using Equation 4.1.10. Edge maps are not stored at each level, but rather new edge pixels found are used to aggregate or grow the existing layer, based on the voting mechanism, which preserves memory. At every accumulator level, the localisation filter is applied to identify duplicate but misaligned edge pixels and avoid double edges. In the process of aggregating layers at every level, new edges are added after passing through the validation function. If no votes are received, then it implies that a new edge was found. The new edge is added to the edge layer if the length is greater than twice the minimum acceptable edge length of four pixels and has no other edge in its vicinity. This eliminates the emergence of spurious edges. In this research, the gradient and smoothing function sets are predefined in the accumulator space. Bilateral and Gaussian filters are used for smoothing, while Sobel, Prewitt, and Scharr gradient operators are used for detecting edge pixels.

The edge growing function R, which is the applied to the vectorised edge layer obtained from Equation 4.1.8 and the vectorised anchor layer, is defined as

$$Rv_t = R(l_{vt}, l_{vt_{anc}}, \omega)) \tag{4.1.11}$$

We make ω a variable to allow improved localisation and merging filters to be passed as parameters.

where l_{v_t} = vectorised output edge layer at level t in the accumulator
$l_{vt_{anc}}$ = vectorised anchor layer that is continually aggregated.
Rv_t = Aggregated layer.

The algorithm of the framework described here is summarised in Algorithm 1. After the final layer is created, the result is a detailed edge map with both object form and detail. Since most of these edges are open curves, edges that are close together are snapped to form closed boundaries where possible. This process is detailed in Section 4.1.6. The final edge map is represented by a graph $G(V_p, E_s)$, where V_p are the vertices that represent the centres of the edge pixels, and E_s is the set of edges that connect the edge pixels.

Algorithm 1 Accumulator Edge Detection Algorithm

Require: $l_{anc} = Null$ ▷ Begins with an empty edge map
Require: $t = 0$
Require: ω ▷ Defined in section 4.1.3, is a variable to Rv_t.
Require: $\sigma step = 1$
Require: $\psi_{step} = 4$ ▷ Step can be varied to reduce accumulator level
Require: x is not $Null$ ▷ input image
  for $\sigma = b$ to a **do**
    $f_t \leftarrow f_g(x, \sigma)$ ▷ Smoothing filter set defined in Section 4.1.1
    for $\psi = m$ to n **do** ▷ Start with highest gradient magnitude threshold
      for g in set g_s **do**
        $g_t \leftarrow g(f_t, \psi)$ ▷ Gradient Operator set defined in Section 4.1.2
        if $l_{vt_{anc}}$ is NULL **then**
          $l_{vt_{anc}} \leftarrow \nu_t(g_t)$
        else
          $l_{vt_{anc}} \leftarrow R(l_{vt}, l_{vt_{anc}}, \omega))$ ▷ Edge growing function Rv_t
        end if
        $t \leftarrow t + 1$
      end for
      $\psi \leftarrow \psi - \psi_{step}$
    end for
    $\sigma \leftarrow \sigma - \sigma_{step}$
  end for
  $l_{en} \leftarrow L(l_{anc}, l_{min})$

4.1.6 Topological Consistency (Cleaning the edge map)

While the accumulator space retrieves maximal edges, over-detecting of edges is inevitable, as with any edge detection process. Both exterior and interior objects are recovered from ABED. The exterior object boundaries define the object form and the interior edges represent the object detail. False edge detection is described in Chapter 5. However, topological editing of the resulting edge map is required before attempting to remove outliers and recover object boundaries. The main editing required is the formation of closed boundary regions from the detected edges. To ensure correctness during vectorisation, a pixel is visited once and assigned to one edge. This implies that junctions, where a pixels are shared between two or more edges, will contain gaps. To recreate junctions and close edges, a direction-constrained snapping filter is used to snap edges based on proximity and predicted edge direction.

4.1.7 Direction constrained Snapping

Edge boundaries retrieved by the tracing algorithm [69] form linear strings of connected edgels based on gradient magnitude, where each edgel has at most a degree of two in the graph $G(V_p, E_s)$. Self snapping of linear edge strings is prohibited. This introduces gaps in the resulting edges. While many edges are retrieved, closing these gaps by snapping to the closest edge pixel results in the formation of false connections. To counter this effect, a direction-constrained sapping approach is developed, where the direction of approach of a given edge chain dictates the direction of snapping to avoid using the closest edge pixel, which is what is implemented in most snapping algorithms. Using the direction of approach produces clean edge maps with vertices of degree of three or more at junctions. Consequently, this introduces information about corners, occlusions, and multilayered surfaces. The result of direction constrained-snapping is illustrated in Figure 42. The direction constrained filter applies a constraint on which direction is allowed for snapping based on the estimated edge approach direction. The direction-constrained filter is defined as:

$$S_{ij} = \begin{cases} 1, & \text{if } \cos(\frac{3\pi}{4}) < f_{d_{ij}} \leq \cos(\pi) \\ 0, & \text{otherwise} \end{cases} \tag{4.1.12}$$

and

$$f_{d_{ij}} = \hat{v_{e_c}} \cdot \frac{(v_p - v_{ij})}{|v_p - v_{ij}|} \tag{4.1.13}$$

where $\hat{v_{e_c}}$ = Estimated edge approach direction

v_{ij} = Candidate pixel to be snapped to v_p

v_p = Candidate pixel to snap v_{ij} to

For example, using Figure 42, suppose there are three detected edges namely e_a, e_b and e_c which belong to set E_s from $G(V_p, E_s)$ and edge snapping and completion is required. Using e_c, the estimated approach direction $\hat{v_{e_c}}$ is computed if the degree of any of the end points of e_c is one. The direction is computed by using the adjacent neighbours of the single degree vertex. From the approach direction, a filter of size $n = [3, 5, 7]$, centred at the first degree vertex, is defined. The filter is constructed by assigning ones to all pixels in the filter neighbourhood whose direction is greater than $\frac{3\pi}{4}$ up to π with the opposite adjacent edge and zeros elsewhere. Snap vertices are then searched for in non-zero filter region based on proximity, with the closest pixel selected. In Figure 42, the red dashed lines are added during snapping, thus completing the edges. The filter size is varied iteratively starting from the least value of the set n.

An example of the snapping and edge completion using satellite data of a tennis court is shown in Figure 43. The red edges are created by the direction-constrained snapping filter. The result is that more closed regions are identified, which paves the way for using the accumulator as a boundary-constrained segmentation method. Furthermore, the three-degree vertices in $G(V_p, E_s)$ are candidates for corners and junctions.

4.2 Non Gradient Based edge detection

While gradient based edge detectors make use of gradient magnitude and direction, some edge detectors rather produce a binary map of the edges. Learning-based edge detectors create binary maps of edges. These edge detectors can be used in the accumulator to leverage on the accumulator's edge growing function. For a given binary map, skeletonisation is required to ensure one pixel width edge map that is vectorised

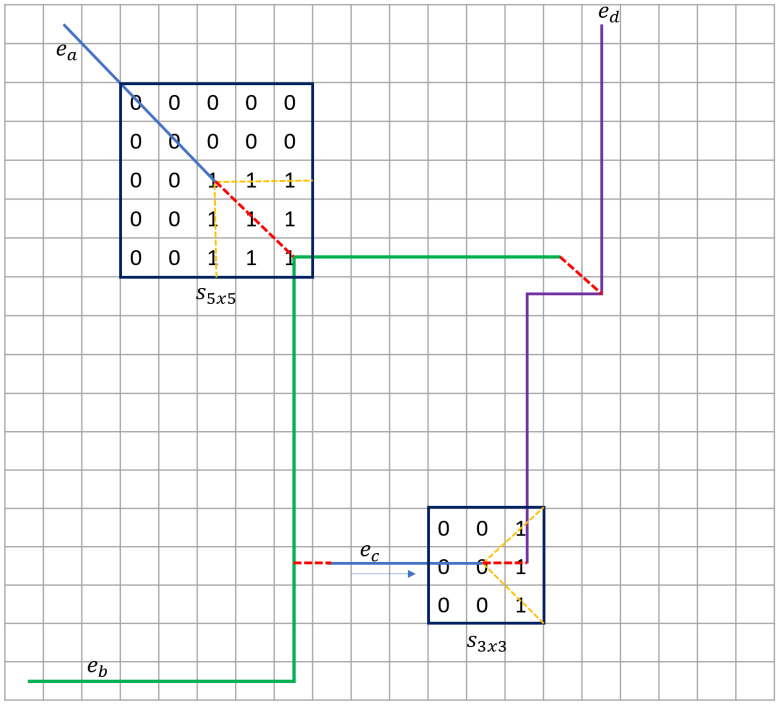


Figure 42: Direction constrained edge completion by snapping using a constrained direction kernel. The red edges have been appended to the existing blue edges using the direction constrained filter.

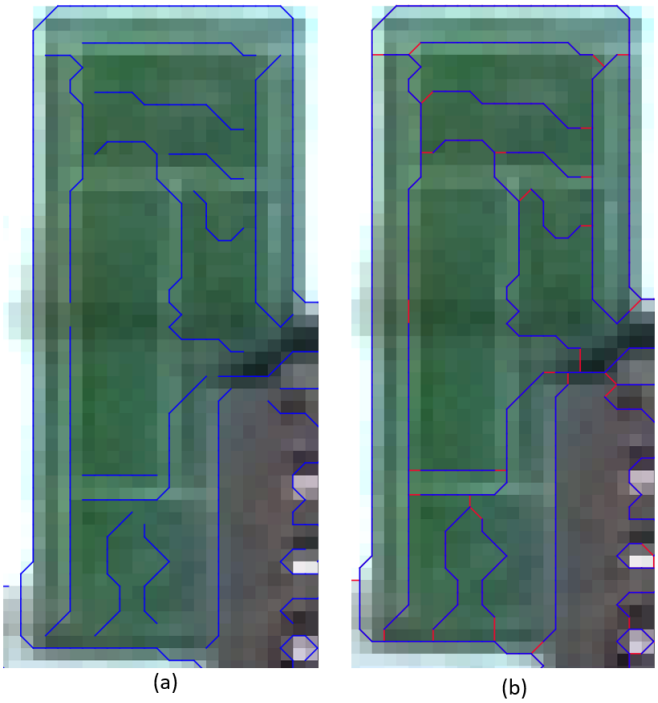


Figure 43: Direction Constrained edge completion by snapping example using a constrained direction kernel. The blue edges have been completed by the red edges in the right image.

into a single boundary map which can then be used in the accumulator. The ability to adapt any edge detector that is gradient-based or produces a binary map shows the genericity of the edge detection framework.

4.3 Discussion

An edge detection framework has been presented. The framework aims to improve edge detection challenges, such as edge fragmentation and missing edges. Sample results of dense edge detection are shown in Figure 44.

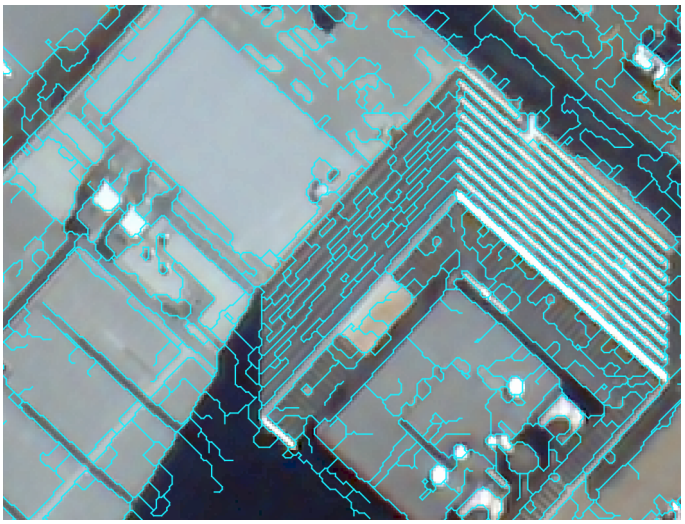


Figure 44: Dense edge map from a satellite image. The detected edges are shown in blue. The recovered object detail in the form of windows is evident on the high rise building and adjacent roofs.

The edge detection framework presented has various advantages. To begin with, the framework aims to be operator-agnostic. This means the results are not solely based on the performance of a single detector but rather allows any gradient based detector to be used in the framework. The framework thus leverages on the strengths of all detectors available in the accumulator. In addition to gradient-based edge detector, any edge detector that produces edges based on some threshold measure can be augmented to the gradient function stack. The same applies to any pre-processing required, which in this case is smoothing. Smoothing function stacks are defined and run on a wide parameter space. The accumulator defines a parameter space for each function to execute. This removes the onus of selecting ideal parameters from the end user. Edges with reduced fragmentation and near invisible edges are automatically detected in the accumulator.

Good edge localisation is maintained using the localisation filter. This solves issues arising from attempts to merge edges from different detectors. By creating a dense edge map, object form and detail are detected. This dense edge map simplifies the boundary reconstruction task. Finally, the edge map can potentially be used for edge-based segmentation, corner, and junction detection. Since the edge map produced is largely connected, the boundary reconstruction task is now achievable through detecting connected line segments.

The one concern with regard to the framework is the run-time requirements. This is a topic that can be explored when optimising the framework. However, with the availability of computing resources, this should not present a bottleneck. Although most edges were detected, a few spurious edges can be seen in Figure 44. However, these edges have been tagged as links and are useful in completing boundaries as discussed in Chapter 6.

Figure 45 shows sample results from ABED on images with dark regions and low contrast. The edges are coloured according to the accumulator levels. As shown in the figure by the different colours, the accumulator detects edges at various levels.

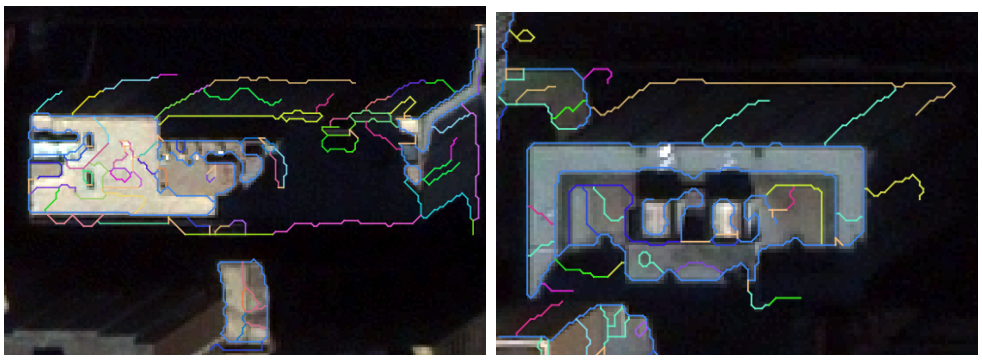


Figure 45: Edge detection in low contrast and shadowy areas. Edges in the dark regions are recovered by ABED in both images. However, the building boundaries are incomplete.

Figure 46 further illustrates the benefits of ABED. Line detection was performed on the edges recovered from the framework presented in this chapter. The results of lines detected using edges from ABED are shown in the bottom right corner, against three popular line detectors, namely Line Segment Detector (LSD), Edge Drawing Lines (EDLines) and Progressive Probabilistic Hough Transform (PPHT) in shadow and low contrast regions. Longer and more lines are detected using the edge map produced by ABED. Detailed results on detector performance are presented in Chapter 7.

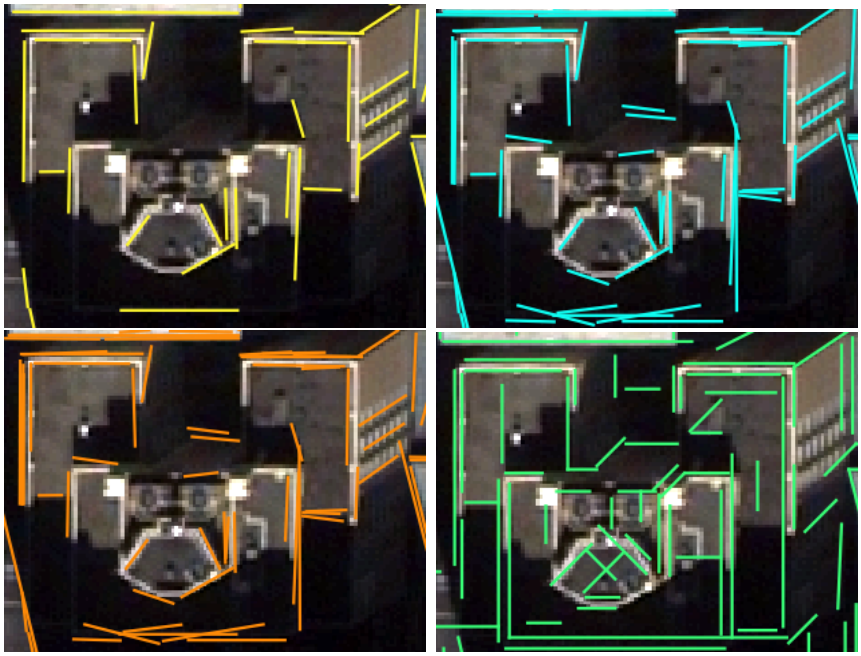


Figure 46: Edge detection in low contrast and shadowy areas. Top left are linear edges detected using LSD, top right are linear edges from EDLines. Bottom Left are linear edges obtained from PPHT and bottom right are the linear edges detected using the ABEDL framework after line detection. Some lines in the dark regions are missing with the first three detectors but are detected in the bottom right image. The bottom right image has reduced fragmentation.

4.4 Conclusion

This chapter presented an edge detection framework that is operator-agnostic for smoothing and gradient operations in detecting edges. By making use of a variable parameter space, maximal boundary retrieval is achieved. This is achieved through an edge growing algorithm that maintains good edge localisation while suppressing noise of varying levels. This results in reduced edge fragmentation. The framework also eliminates the need for parameter tuning for different operators. The effects of shadows and low contrast are greatly reduced. Finally, the dense edge map that captures object form and detail is created. The dense edge map opens up the possibility of attempting full reconstruction of buildings from edge features, which is an area that has not been fully explored. This is discussed in Chapter 6.

5 Boundary Detection and Reconstruction

The previous chapter presented the ABED framework that recovered dense edges of objects. These edges capture both object form and detail. These edges are represented as a set of chained vector lines or linestrings that connect edge pixels. From this dense edge map, the aim is to recover interior and exterior object boundaries that follow object form and detail. This chapter presents a boundary recovery method in Section 5.3, which traverses the dense edge network to recover boundaries with correct orientations. These boundaries are then connected to form longer boundaries that allow for building boundary shape reconstruction. The boundaries of the objects are recovered by implementing a line and curve retrieval algorithm from the produced edge map. Furthermore, the problem of recovering linear versus non-linear edges in images is solved by an analysis of line behaviour when projected onto an image along any given direction. The results of this analysis are then used to recover the true edge orientation from a given edge map. Sections 5.4 presents a method to enhance and retrieve longer boundaries followed by outlier detection in 5.5. Finally, a discussion is given in Section 5.6. Boundary Detection and Reconstruction are the first part of the building boundary shape reconstruction framework (ABEDL). First, we discuss object representation and the boundaries that are being recovered in Section 5.1 and some current challenges in boundary recovery in Section 5.2.

5.1 Object Boundary Representation

Object boundaries can be graphically represented in vector form. In 2D graphics, any object shape or form can be modelled using a combination of lines and curves of varying orders. However, rendering curves is achieved by chaining small line segments with smoothly changing direction. Put differently, curves and lines can be defined as a series of connected linear segments of varying lengths and orientations. This is illustrated in Figure 47. The circular part is a series of connected small line segments. Consequently, the task of boundary reconstruction from the edge map created in the previous chapter is reduced to recovering a series of connected lines, forming linear and curved boundaries. In other words, boundary recovery can be defined as the task of retrieving

a set of connected lines of varying length and slope in 2D that form closed boundaries in and around objects. The final boundaries of the individual objects are then recovered from these connected lines. Each of the steps to recover the true boundary of the object is explained in detail in this chapter. Curves are detected by decomposing the curve into piecewise-linear segments, rather than by defining implicit curve equations. This avoids the need to define a model to use for curve fitting.

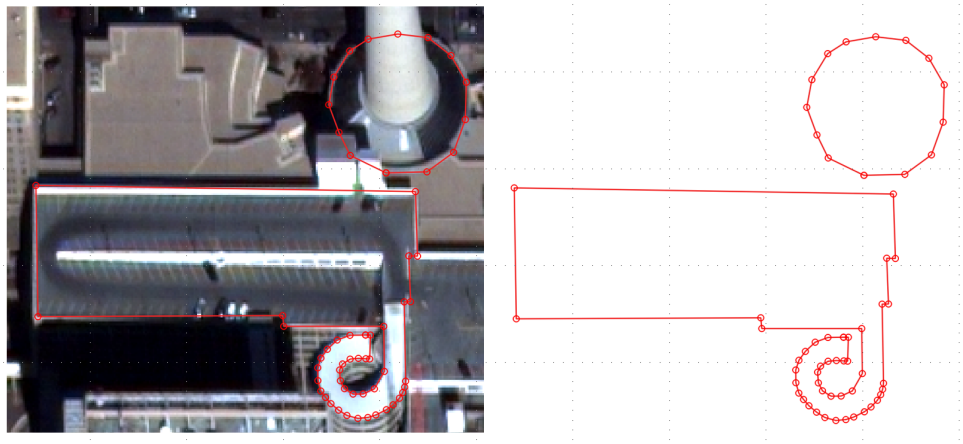


Figure 47: Curve Representation using vectors.The building boundary in red is a set of connected piecewise linear segments. Curves are represented using connected linear segments as shown in red on the right image.

5.2 Challenges in Boundary Recovery

Satellite images present various challenges when recovering object edges. Satellite image resolution is still limited to $0.3m$. The reduced resolution results in the pixel mixing problem, where radiometric responses of multiple objects are mixed in one pixel [235]. Object boundaries are sharp features that occur over a very small range, which is less than the imaging resolution. Consequently, edges are not captured as distinct features, but rather occur over a range of pixels. Most edge detectors are able to compute the central pixels where the highest gradient transition occurs to indicate the best locality of the edge [3, 1]. However, this becomes more challenging at object corners, where sharp corners, which are point features, are blurred due to increased pixel mixing and imaging resolution that does not capture point features distinctly. The regular grid nature of pixels adds to the problem in that the dominant reflectance is often captured. These phenomena often lead to edge fragmentation during edge detection at corners, where

edge end points do not terminate at object corners [6, 78]. Smaller sized object shapes are not captured accurately as a result of reduced image resolution. For example, a chimney may appear as a cluster of three pixels. Furthermore, the limited radiometric resolution and sensor dynamic range of satellites reduce edge fidelity between objects of similar reflectance. This lack of huge contrast variations results in missing edges [210, 99]. Furthermore, satellites are used to acquire large areas, and thus the images are affected more by illumination variance and shadows. As a result, recovering full unfragmented boundaries from satellite images remains a challenge.

In addition to the challenges mentioned above, building boundary recovery from edge pixels has been treated as an optimisation problem [92, 66]. From pixels recovered from any edge detection process, boundaries are recovered by fitting linear segments to pixels that best describe a line or a curve. Most boundary recovery methods or edge detectors benchmark their performance based on the number of linear segments detected [4, 3, 185]. Linear segment retrieval based solely on optimisation has some limitations that hinder the successful retrieval of all edges. Three major drawbacks exist when using line fitting as a boundary recovery method. Firstly, while a line fit is always guaranteed for a given point cluster within a defined threshold, it does not imply that the boundary detected as a line is accurate. This is illustrated in Figure 48, where the red line's curved form is lost and the edge is fragmented in green as a result of line fitting. Smooth transitioning curves are ultimately detected as disjoint linear segments. Consequently, information about the curved nature of the boundary is lost. Attempting to fit lines locally on edge pixels adds to line fragmentation and incorrect line orientation in the presence of noise and line junctions. To mitigate this, an incremental boundary retrieval algorithm is proposed that attempts line fitting with constraints to reduce fragmentation and inaccurate line orientation.

Secondly, fitting a line within a given threshold at times introduces a rotational error in the recovered lines. This error is often introduced by object corner points that fall within the fitting threshold for a given point cluster. This is illustrated in Figures 49 and 50 where circled lines appear slightly rotated compared to the actual line orientation. A line fitting algorithm that minimises these rotational errors is proposed in Section 5.3.5

Thirdly, shadows introduce false edges. Dark areas covered by shadows will have their

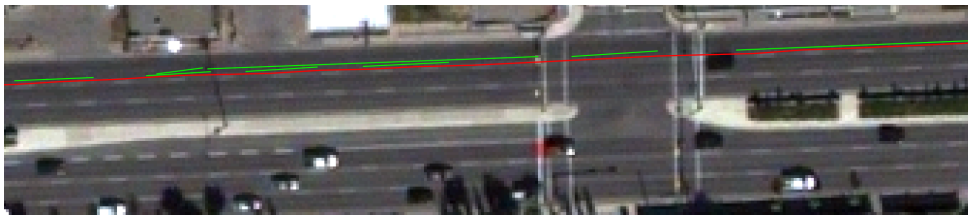


Figure 48: Edge Fragmentation during line fitting. The red line shows the edge from a road that is a smooth curve. However, using EDLines to detect the same edge results in fragmentation due to the straight line fitting. The curved nature of the road is not preserved.

Figure 49: Errors introduced as a result of line fitting. While all roof edges are supposed to be perpendicular, some rotations can be seen with some of the edges, resulting in non-parallel and non-perpendicular edges circled in red. Edges in the left image were recovered using LSD and the second set of edges on the right were recovered with EDLines.

Figure 50: Errors introduced as a result of line fitting. Edges circled in red show random rotation when compared to the expected direction. Edges in the image above were recovered using EDLines.

own boundary. Buildings under these shadows are consequently missed. Other buildings that are partly covered by shadows are partially detected. This effect is illustrated in Figure 51. The left image shows edges detected by LSD and right side detected by EDLines. The red circled buildings show the partial detection as a result of shadows. Although these false edges are unavoidable, ABED has the ability to recover the invisible part of the building boundary.

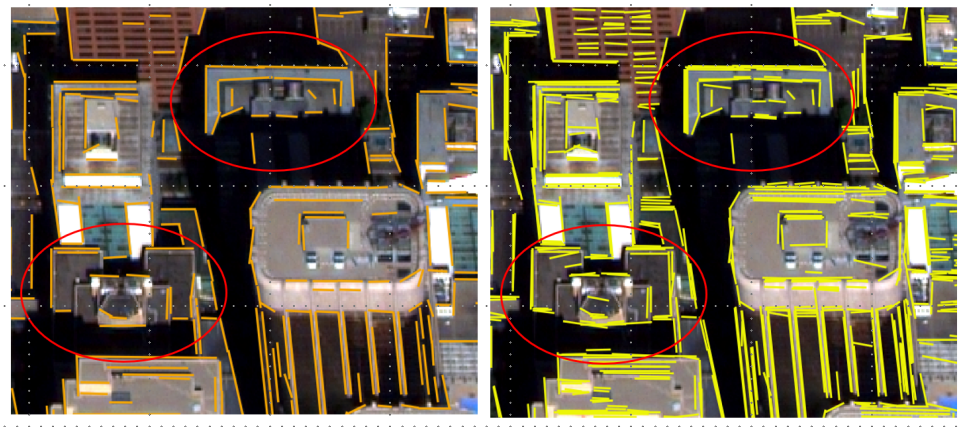


Figure 51: Errors introduced as a result of shadows. False object edges introduced as a result of shadows are circled in red. Edges in the left image in orange were recovered using LSD and the second set of edges on the right in yellow were recovered with EDLines.

This chapter presents a new line and curve recovery method that aims to improve on the issues highlighted in this section. The next section proposes a new method for recovering lines from detected edges while reducing fragmentation and improving line orientation. This method recovers these boundaries under varying image conditions. The novelty of the proposed boundary recovery lies in the line recovery method. Most methods treat lines as independent entities. When fitting lines independently, local information is only considered. For example, a junction point may be shifted after line fitting. This results in line end-point mismatches and disconnected lines. The method proposed here uses global information to fit the lines. If junction points are encountered during line fitting, the orientation of lines incident to that junction point are introduced as constraints. This preserves the orientation of lines globally. This is achieved by treating the edges as a network of connected lines in a graph. The line recovery method presented here is the backbone of ABEDL.

5.3 Boundary Detection and Reconstruction workflow

The boundary detection and reconstruction workflow presented in this chapter takes as input the dense edge map created by the ABED framework. The boundary recovery task is reduced to finding a set of connected piecewise-linear segments. The edge map is stored as a graph $G(V_p, E_s)$. Consequently, lines are recovered by traversing the entire graph using a breath-first search strategy. Junction points are introduced as constraints during line search. The boundary recovery workflow consists of five steps as illustrated in Figure 52.

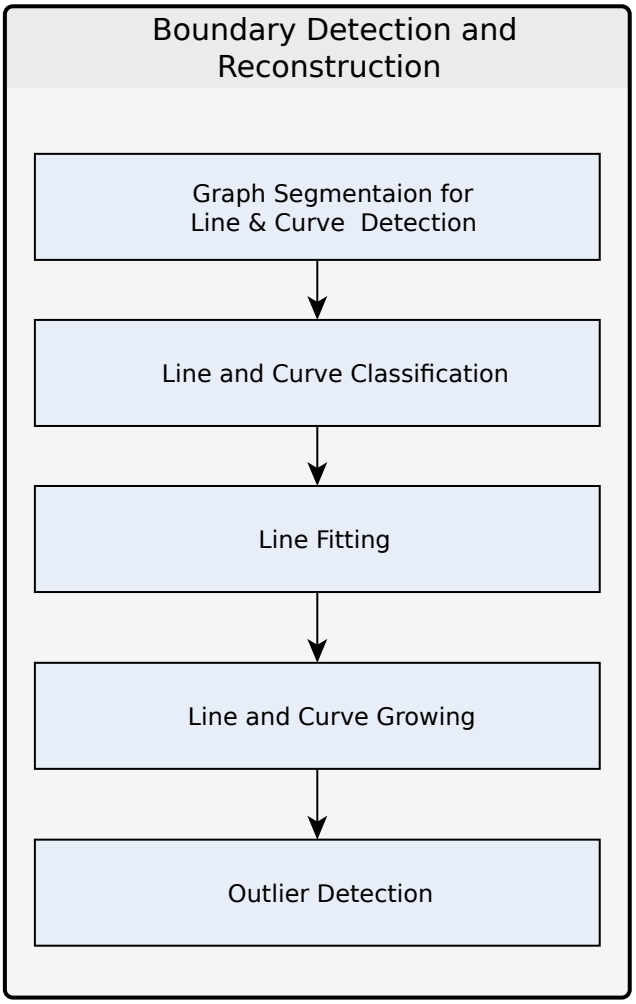


Figure 52: The line Detection workflow proposed in this research. The workflow is comprised of five steps.

1. Graph Segmentation (Line Detection): In this step, initial line segments are detected based on four directions.

2. Line and Curve Detection and Classification: Line groupings, termed linesets, that belong to a single curve are identified by graph traversal and classified into four classes defined in Section 5.3.4.

3. Line Fitting: Lines are fit to the grouped linesets

4. Line aggregation: From the lines that are fit, line aggregation using adjacent neighbours is carried out to extend lines and curves.

5. Outlier detection: From the recovered lines and curve classes, an initial set of outliers is detected.

The following sections provide a detailed description of these steps.

5.3.1 Graph Segmentation (Line and Curve Detection)

. When projecting lines onto an image grid, only four directions are used. These directions are the horizontal x axis, the vertical y axis and two opposite 45^o diagonals. These directions are represented by the set D_s, where $D_s = [0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}]$, with the x-axis as reference. The true line direction in an image is preserved without distortions along these four directions. For any other direction, lines do not appear straight, but rather as small lines of direction D_s with periodic stepping or stair-stepped appearance. This stepping is strictly increasing or decreasing for a straight line. This is illustrated in Figure 53, image (a), where the blue line segments with direction D_s, are connected by the red lines that are the steps.

[223] presented a line detection approach based on the stepping property. Cho et al. [223] refers to the blue line segments as linelets in pixel space. By grouping linelets with similar properties, lines were recovered. However, the method developed is limited to straight lines. This section extends this property to incorporate curves. As shown in the results section, linelet grouping suffers from fragmentation due to pixel mixing and the previously mentioned fact that edges in satellite images appear over a wider region, which interrupts the line grouping properties.

Similarly, curves and circles are also projected with stair stepping. However, there are two differences between the straight line stepping and curve stepping. For linear segments,

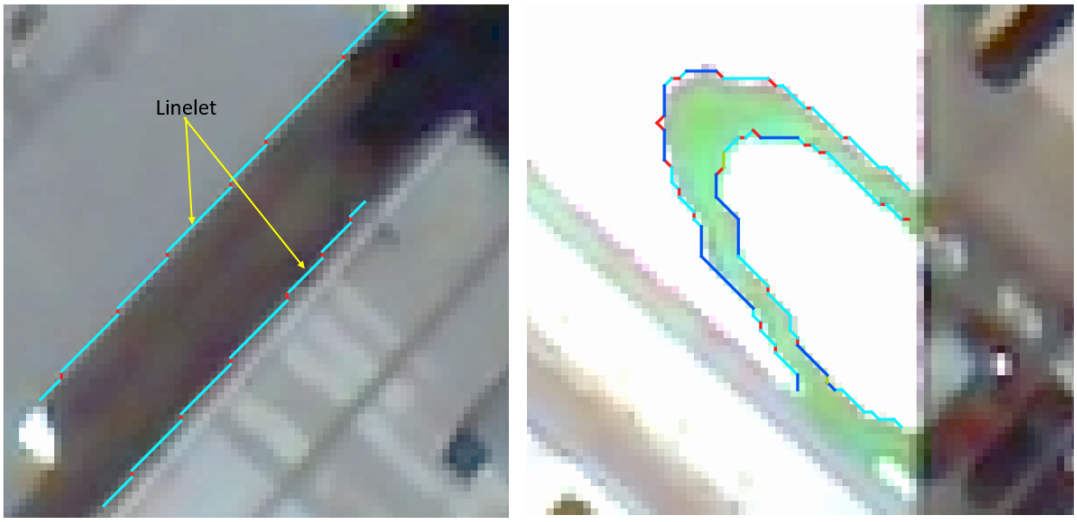


Figure 53: Edge behaviour after projecting a line and curve onto an image grid. Image (a) shows the resulting stepping of a line projected onto an image in red, and image (b) shows the stepping of curves projected onto an image. Cho et al. [223] refers to the blue line connected by the red steps as linelets in pixel space.

the steps connect linelets of one constant direction. For curves, the steps connect linelets with different orientations. However, the maximum difference in the orientation of the linelets for a curve is set to 45^o. The stepping behaviour for curves is illustrated in image (b) of Figure 53. By using this property during graph traversal, curves are detected. In this research, linelets are grouped into sets of straight lines or curves, termed linesets. A lineset is a grouping of linelets that belong to the same straight line or curve.

The stepping is further illustrated in Figure 54, where the lines ls_1, ls_2 ,ls_3 and ls_4 with directions from D_s, the line orientation and form are preserved. However, along other directions, ls_5 and ls_6 of direction 113.268^o and 76.964^o respectively, the stair stepped effect is visible.

Put differently, a periodic horizontal displacement is often followed by a vertical step when representing lines of arbitrary directions with the exception of direction set D_s. This is illustrated in Figure 55 of a high rise building that has projective distortion. However, it is observed that in pixel space, any other line direction other than the four main directions mentioned can be represented by two of the four main directions from D_s. Using these characteristics of line and curve behaviour when projected onto an image, a line and curve recovery method is developed.

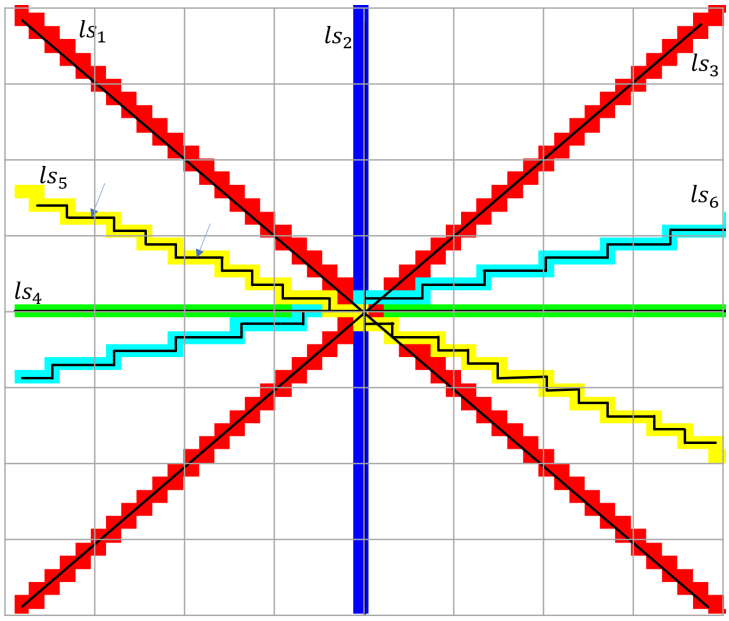


Figure 54: Edge behaviour after projecting a line to an image grid. Lines ls_1, ls_2 ,ls_3 and ls_4 with direction D_s have undistorted lines without stair stepping while ls_5 and ls_6 are characterised by a strictly increasing or decreasing stair stepping

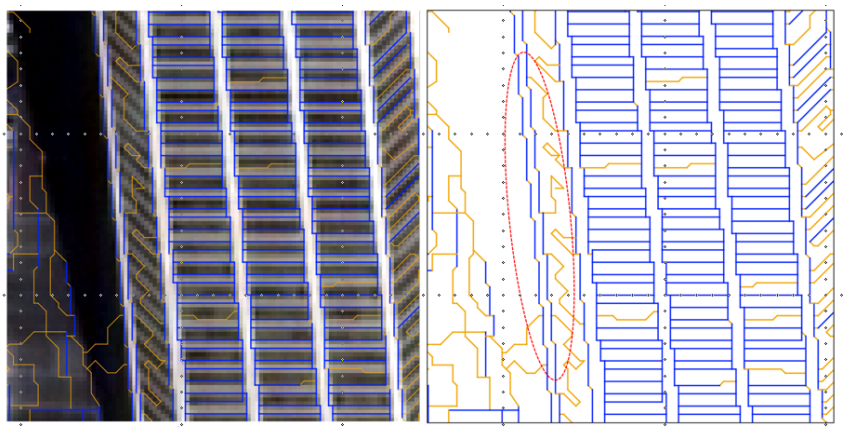


Figure 55: Example stepping effect for lines that do have a direction in D_s along the building vertical edges. The blue lines are straight lines with direction not from D_s that appear with stair stepping.

5.3.2 Line Detection

The line detection and reconstruction method presented uses the stair stepping property. Lines are detected in two separate steps. The first set of line segments detected have directions from set D_s. The second set of lines are directions that do not belong to D_s. To detect both sets of lines, an undirected graph G_{ac} is created from the edge map that is output from ABED. Graph G_{ac} is defined as:

$$G_{ac} = G(V_p, E_s) \tag{5.3.1}$$

where V_p = Vertices of the graph representing each edge pixel
E_s = Edgels connecting adjacent edge pixels

The first step is to detect lines that belong to D_s in G_{ac}. In G_{ac}, the edges E_s, are initially connecting adjacent edge pixels. These small connections are called edgels or unit edges. A segmentation function is defined that outputs lines belonging to D_s in G_{ac}. From the edgels connecting adjacent edge pixels, the segmentation merges lines along the four directions D_s by placing a collinearity constraint on the four directions. The constraint is that second-degree vertices on a line of direction D_s only exist at segment end points. Graph G_{ac} is traversed while searching for collinear edgels along any direction in D_s. For any set of collinear edgels detected, a new segment replaces the edgels. Three other constraints are placed on the graph G_{ac}:

1. Junctions and corners are assumed where the degree of the vertex is greater than two.
2. A segment is allowed to have a collinear vertex only if the degree of the vertex is greater than two.
3. A segment with a direction in the set D_s is a separate segment

From these constraints, the segmentation function is defined as:

$$S_G = \{E_{ls} \mid D_{ls} = D_s \mid degree(V_{ls}) < 3 \mid E_{ls} \not\subset E_s \mid V_{ls} \subset V_p\} \tag{5.3.2}$$

where V_p = Vertices of the graph representing each pixel

E_{ls} = Line segments with direction from D_s

D_{ls} = Direction of line with in D_s

The segmented graph is comprised of a list of edge segments that have four main properties:

1. Each edge string E_{ls} has a direction from D_s.
2. No collinear vertices exist in the interior of an edge E_{ls} with direction D_s except at line junctions.
3. Only vertices with a degree greater than two are allowed to exist in the interior of a straight segment with direction D_s.

The segmentation function creates a new list of edges and a new graph $G_{ls} = G(V_{ps}, E_{ls})$ with new line segments that strictly follow D_s. V_{ps} are the vertices of the new segments E_{ls}. E_{ls} still contains some unit edges. These are edgels that have not been merged into larger segments and are mostly step segments. Figures 56 and 57 show segmentation results from the given edgels in the left image and the merged segments based on the constraints defined in the segmentation function in the right image.

The second step is to detect candidate lines that are not from the set D_s. A breath search function F_{gs}, traverses G_{ls} and searches for periodic steps that are strictly increasing or decreasing, connecting any initially detected line segments. The function merges all linelets with periodic stepping, thus creating linesets. The function F_{gs} finds connected components in graph G_{ls}, where edgels representing the steps, are the connecting entity. Figure 58 shows the detected lines with direction D_s in blue. Figure 59 shows the sample results on the connected components recovered by the breadth first traversal function. Each detected component has a unique colour. Lines with stair stepping are detected as a connected component. The results of the connected components form the second set of lines that do not belong to D_s. Each component is referred to as a lineset. Lineset detection is performed on the vectorised edge map as opposed to pixels.

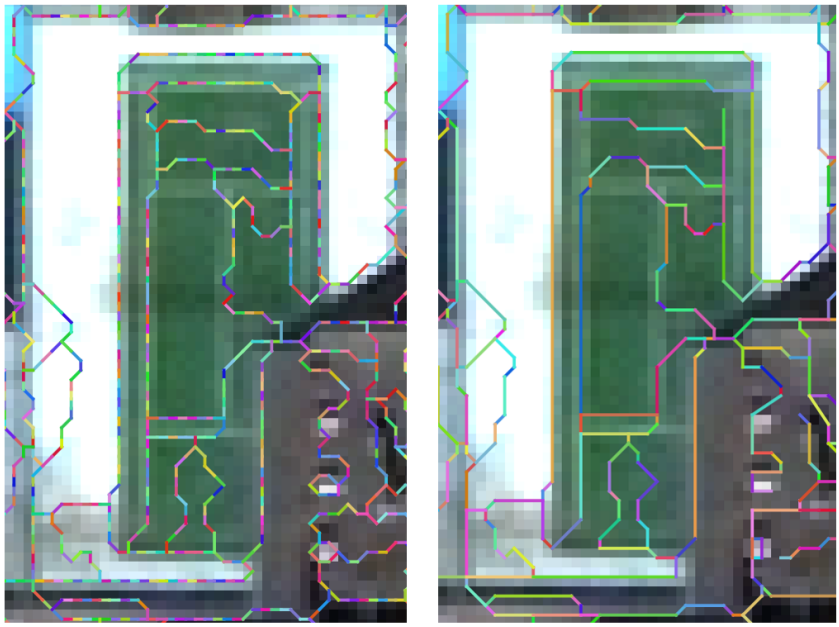


Figure 56: Left image shows the edgels from the edge map on a 0.5m satellite image of a tennis court. The right image shows the segmented edges. Longer segments are created from the segmentation.

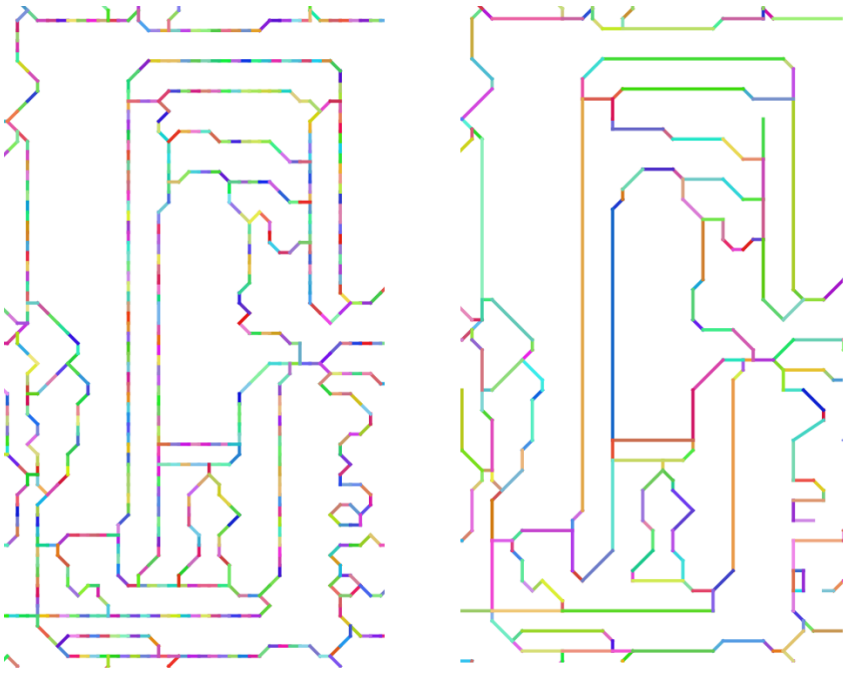


Figure 57: Graph based edge segmentation. Left image shows the edgels on Figure 56 without the background image. The right image shows the segmented edges. Longer segments are created from the segmentation

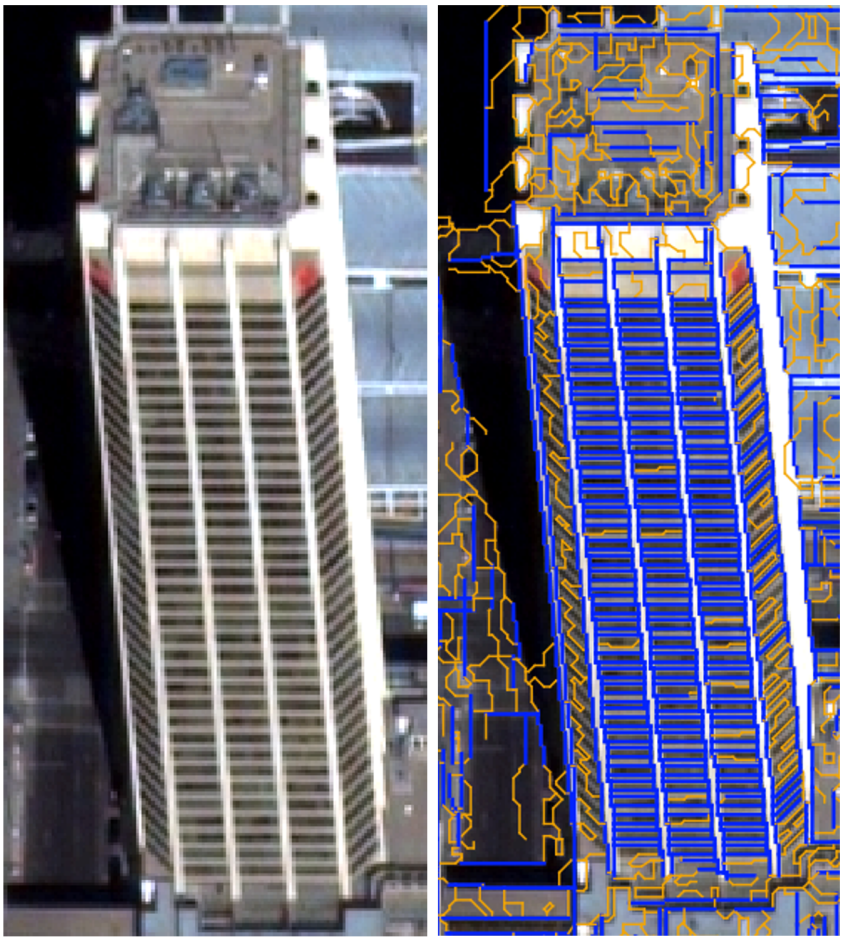


Figure 58: Detection of line candidates based on line step detection. Blue lines are the initially detected lines. However, these lines have a constant stepping along the building since the building vertical edge direction does not belong to D_s.

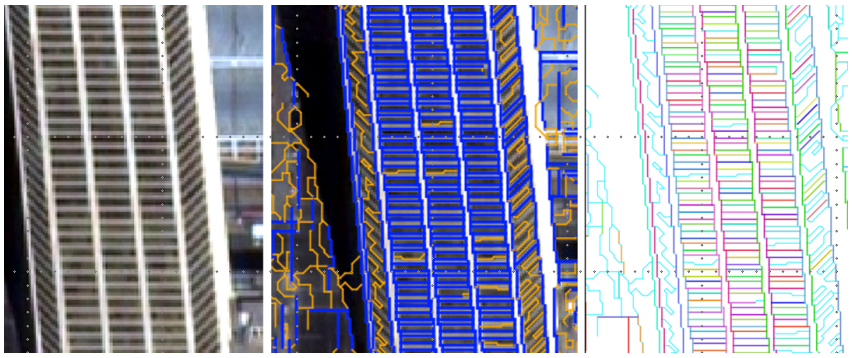


Figure 59: Sample results of connected components search for lines using F_{gs}. The middle image shows the detected lines with direction D_s. Third image shows detected components in different colors connected by the steps.

From the detected linesets, initial lines are recovered by connecting the end points of the linesets. Sample results of detected lines are shown in Figure 60 where the lines are recovered by end point connection. The desirable property is that minimisation is not required at this stage. These initial line segments are used in the subsequent steps for curve detection. However, minimisation is recommended as a final step to refine line orientation. Line connectivity from dense edge detection is maintained during the line recovery steps.

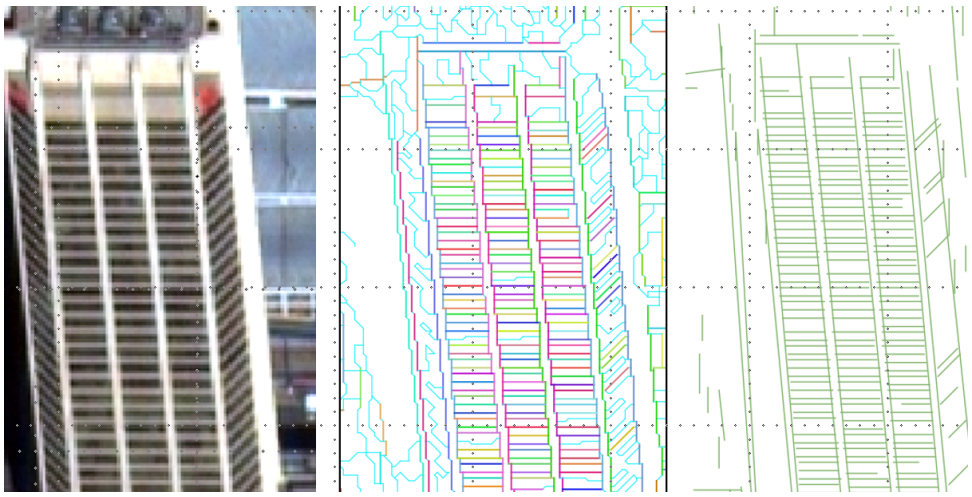


Figure 60: Example of recovered lines without minimisation. The lines in green in the rightmost image are recovered from the classified components in the middle image. The stair steps have been used to recover line with $d_l \neq D_s$ without line fitting. The parallel nature of the lines is preserved.

5.3.3 Curve Detection

Connecting the end points of a lineset gives the initial lines in the graph. Curves are implicitly recovered in the last step. By traversing the graph G_{ac}, and finding connected lines with smoothly changing orientations, smooth curves are recovered. The lines used are the initial segments recovered from connecting lineset endpoints. To recover curves, linesets are further merged according to the constraint that the difference in orientation should not be greater than $\frac{\pi}{4}$. The merged linesets form new linesets that represent curves.

To begin with, a curve is defined as a series of piecewise continuous linear segments connected to form a chain of lines with smoothly changing directions. Looking at set D_S, the direction are separated by $\frac{\pi}{4}$. The maximum angle allowed between consecutive

connected lines is set to $\frac{\pi}{4}$. In other words, any three linear segments where the difference in direction between consecutive segments is at most $\frac{\pi}{4}$ are considered a curve. A depth first search is applied on G_{ls} to connect lines that form curves. By discovering connected components based on this constraint, the curves are recovered. This is depicted in the Figure 61 below.

$$C_s = \{E_{cs} \mid E_{cs} \subset E_{ls}\} \tag{5.3.3}$$

where E_{cs} = Connected Components (edges) of the curve in graph G_{ls}

C_s = is any given curve

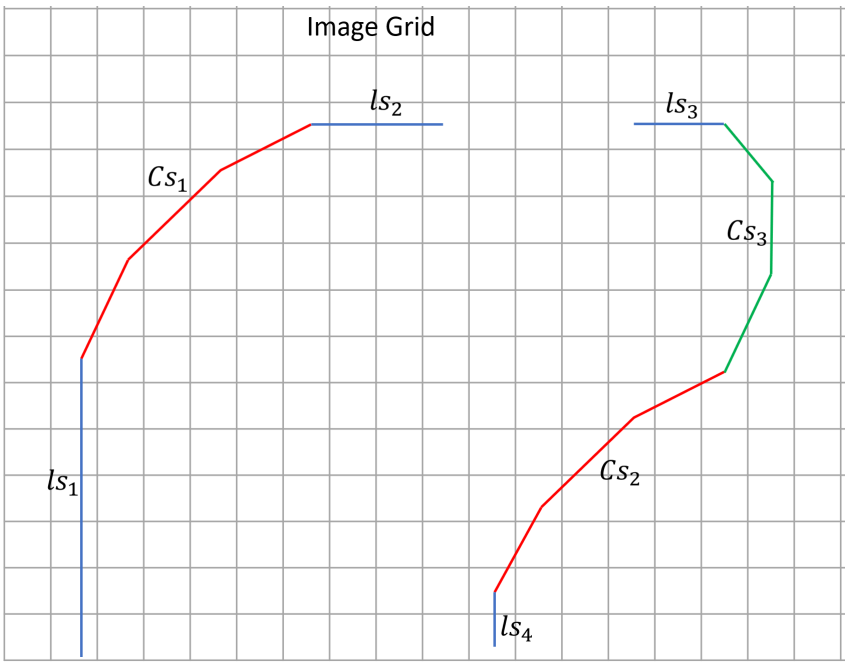


Figure 61: Definition of a curve given a set of linesets. The red and green segments represent the definition of a curve for this research. $C_{s_1}, C_{s_2}, C_{s_3}$ are separate curves composed of connected lines with smoothly changing direction. l_{s_n} are detected line segments. The change in direction should be strictly clockwise or anticlockwise for a single curve.

The curve recovery equation is thus defined as:

$$C_s = \{E_{cs} \mid D(E_{n+1}) - D(E_n) < \frac{\pi}{4} \mid degree(V_{cs}) > 1 \mid E_{cs} \subset E_{ls} \mid V_{cs} \subset V_p\} \tag{5.3.4}$$

where E_{cs} = Connected Components (edges) of the curve in G_{ls}

C_s = is any given curve

E_n = is any given line segment in C_s from the previous line detection step

The detection of a curve is defined as the discovery of piecewise connected segments with smoothly changing direction. The change in direction should not exceed $\frac{\pi}{4}$ but always non zero. Since our edges are represented as a graph, detection of curves is reduced to the discovery of connected components of linesets that meet the criteria of a curve defined in this section. Example curves detected are shown in Figure 62

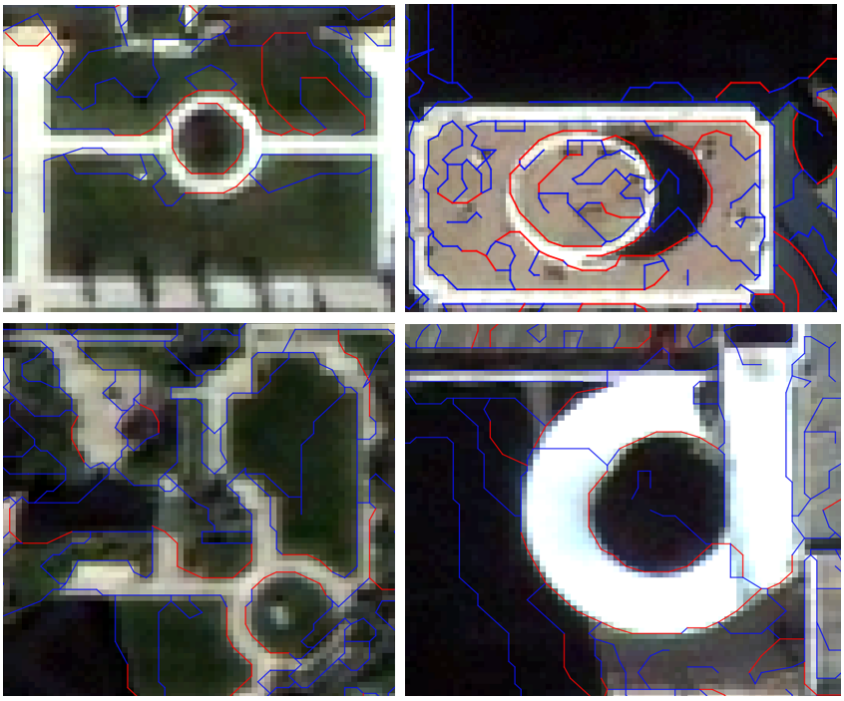


Figure 62: Examples of detected curves in the image in red. The red edge strings are used as input in shape recovery. Spurious edges are used as links or removed during reconstruction.

Figure 62 illustrates the results obtained from the curve detection function. Various types of curves are detected, some which are part of ellipses or circles. This detection simplifies the subsequent shape recovery and reconstruction step.

These linesets or curves are sometimes interrupted and fragmented as a result of small errors or small objects that introduce spiking in the edge map. By removing these errors, lines and curves grow, resulting in longer curve and line segments. This is discussed in Section 5.4. The next section discusses line classification is discussed.

5.3.4 Line and Curve classification

The results of the previous steps are connected components of either lines or curves. Each detected component is classified into four classes, namely a straight line section that belongs to D_s, a straight segment where direction of the segment $d_l \neq D_s$, a curve section and a non-classified linear segment which are classified as links. While the first class is detected by the first set of constraints given in Section 5.3.2, the second class involves identifying the stair stepped nature of the segment. If a repetitive pattern is identified in the stair stepping then the segment is considered linear but with $d_l \neq D_s$ otherwise it is classified as a smooth curve if detected as such in the curve recovery step. Unit edges that remain are classified as links.

Identifying lines with a near repetitive stair stepped appearance will solve two main issues over existing methods. Firstly, the selection of points for line fitting is globalised by connected linesets as opposed to a local search for point clusters that fit a line, which often introduces spurious points that affect true line orientation recovery. Secondly, fitting becomes optional in the stair stepped components. Connecting the end points of these components recovers the line. However, verification by line fitting with orientation correction is recommended. Line orientation correction is discussed in more detail in Section 5.3.5.

5.3.5 Line Fitting

Although lines can be recovered without line fitting, an optional line fitting step is detailed in this section to verify the lines recovered. The purpose of this step is to have an independent line verification algorithm on the recovered lines. The line-fitting method proposed here can be used to verify and correct line orientations. The proposed line fitting also reduces the errors introduced by current line fitting methods where there is no prior knowledge of edge characteristics or attributes. Points are rather clustered based on gradient orientation without knowledge of junction points or adjacent lines. This tends to introduce fragmentation or orientation errors. To counter this, a new line fitting algorithm is proposed [98]. Since a connected graph of edge points was created as output in section 5.3.2, knowledge of junction points, corners and adjacent line is

contained in the graph.

Most line fitting methods minimise the point to line distance d for any given point on the line using Equation 5.3.5.

$$f(t,\theta) = \arg\min_{\theta} \sum_{i=1}^{n} d(t_i,\theta) \tag{5.3.5}$$

where θ = line parameters

t_i = line point i

The line is then obtained using the parameters θ to retrieve the line ι_a.

$$\iota_a = L(t,\theta) \tag{5.3.6}$$

where L = the line retrieval function

t_n = point set of size n used in computing θ

The Lines detected with Equation 5.3.5 using image edge points have a slight error in the direction of the line. This is as a result of object corners and junctions. Points at the corners of the object at times introduce a slight rotation in the resulting line. This is because most line-fitting methods distribute the errors equally using a fixed distance function. For example, suppose that a line is fit with a maximum error threshold of ε, the line ι_a is obtained since the pixels on the corner edge fall within this threshold instead of ι_b as illustrated in Figure 63.

This rotational error can result in junction points shifting with the line. To correct for this error, the pixel errors at the end of the line need to be excluded in the fitting. This is achieved by performing an iterative line fit from the midpoint of the line going outwards towards the line endpoints. This allows less rotational freedom of the line in the final result. Similarly in Figure 64. The desired line is either ι_b or ι_c and not ι_a which has a directional error compared to the true direction.

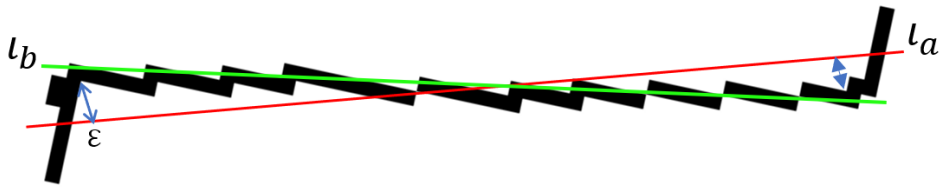


Figure 63: Rotational line fitting error introduced by corner edge pixels that fall within error threshold ε defined for the line ι_a. The desired line is ι_b.

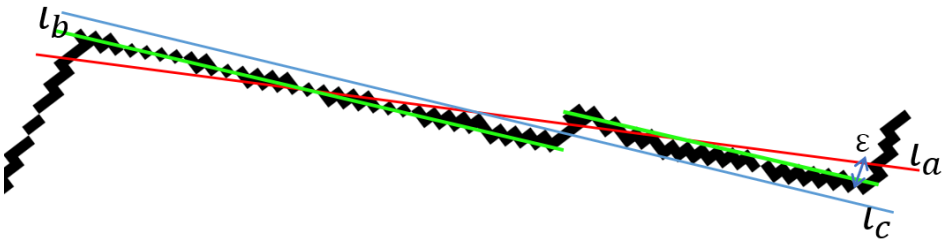


Figure 64: Rotational line fitting error introduced by corner edge pixels that fall within error threshold ε defined for the line ι_a. The desired line is ι_b or ι_c. As shown with ι_b and ι_c, line direction is maintained.

To correct for orientation errors, an initial line is fit using Equation 5.3.5. This gives us the line that might have a rotational error as a result of corner points. After obtaining this initial line, we now correct for the rotational error if it exists. This is achieved by reducing the line points at either ends by a small percentage g and then computing $f(t, \theta_g)$. Line points are created by densifying lineset points along the lineset segments. The results are then used to compute δr, the difference in direction between the directions computed by $f(t, \theta)$ and $f(t, \theta_g)$. If this exceeds a set threshold ϵ, then parameters for $f(t, \theta_g)$ are adopted otherwise $f(t, \theta)$ is kept as the correct parameters for the line. Since all line connectivity is preserved, if a junction point exists on the line, its position is fixed during line fitting. This preserves the orientation of other lines incident to the junction and introduces global constraints to line fitting.

This calculation can be improved by iteratively varying the value of g while δr remains less than ϵ with the first iteration as the reference direction. The initial iteration must contain at least half the line points used to obtain $f(t, \theta)$, centred around the line midpoint. Consequently, the parameters calculated from the last iteration with

δr below ϵ are adapted as the correct parameters. This process is depicted in Figure 65 below where the last iteration before the directional error threshold is exceeded is selected. ϵ value was set to 2.5^o in this research.

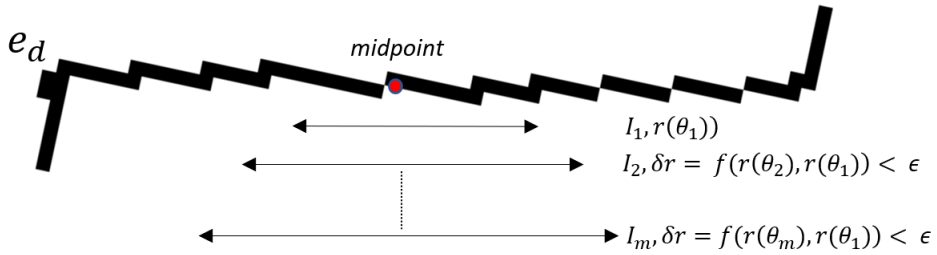


Figure 65: Line fitting with directional error control. Fitting starts from the midsection of the line going outwards maintaining a line direction variation δr that is less than a threshold ϵ at each iteration I_m. The last line before exceeding the error threshold is selected.

This algorithm is optionally used to refine the line direction for all lines obtained during graph segmentation explained in section 5.3.1. In addition, any least-squares line fitting methods may be improved by implementing this line fitting and refinement algorithm.

5.4 Line and Curve Growing

Line growing, which is the process of recovering longer unbroken line segments, is achieved implicitly. Effects such as small spiking and stepping of adjacent pixels are often noticed in any edge detection process, as shown in Figure 66 in red. This is as a result of the distortion that occurs when projecting arbitrary lines onto an image grid, coupled with image resolution, noise or small objects. These small artefacts, namely spiking and stepping, can be cleaned out to improve topological consistency of the resulting detected boundaries. In addition, this spiking has a negative impact on the line fitting process as they contribute to the rotational error discussed in section 5.2. To account for this, some constraints are defined on the graph G_{ls} through a consistency filter. The first constraint is that consecutive unit edges that are not collinear and connected by a second-degree vertex are not permitted. The second constraint is that connected non-collinear unit

edges can be replaced with a straight line if the vertices are within half a pixel length of the resulting line. Applying these two constraints results in cleaner topology as shown in Figure 66. In Figure 66 the input segments in red are the leftmost column. The middle column shows the results in green overlaid with the input edges after applying the constraints on G_{ls}, and last column shows the cleaned output lines. Cleaning these errors ultimately allows longer line retrieval (line and curve growing). The results are cleaner and longer edges that allow for more accurate line and curve recovery.

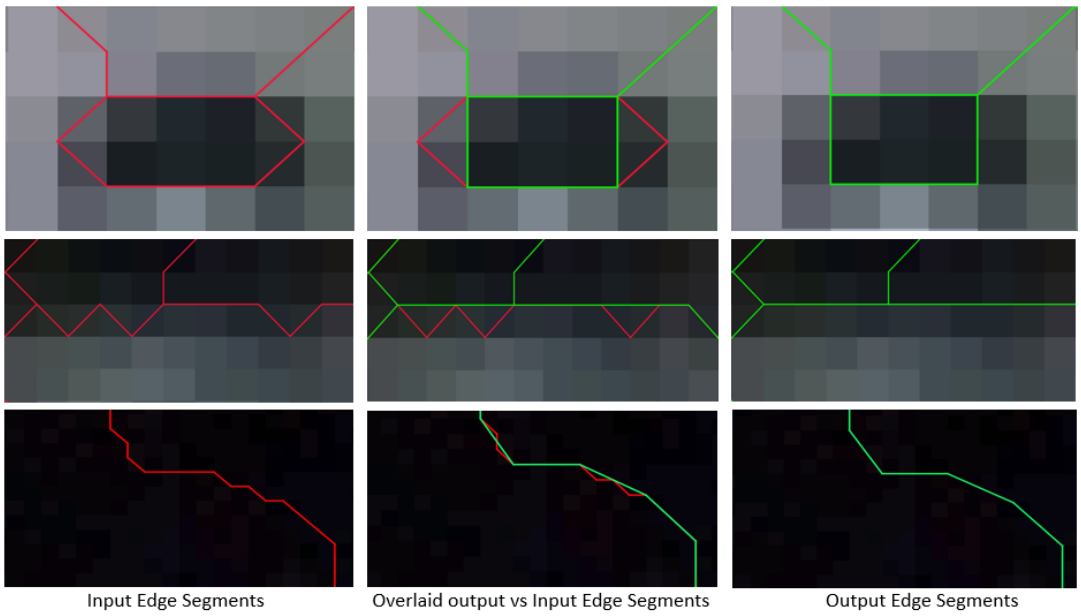


Figure 66: Topological consistency filter. Input segments in red in the leftmost column that have spikes and kinks. The middle column shows the results in green after applying a topological consistency filter overlaid with input edges (red) and the last column shows the output cleaned edges. The result is cleaner long edges that will allow more accurate curve and line fitting

5.5 Outlier Detection

Not all edges recovered by the accumulator represent true object boundaries or object details. These edges need to be identified and then assessed to determine whether they are weak boundaries or spurious edges. Spurious edges are usually classified as links in the line classification stage. Since classification is purely based on whether a line is a curve or not, unclassified edges are not entirely spurious. To identify the final true edges, unclassified edges are further classified based on edge strength and connectivity. Weak edges are generally added at the bottom of the accumulator stack so they have

a low accumulator layer id. Weak edges are further assessed based on connectivity. If in graph $G_{ls} = G(V_p, E_{ls})$, a weak unclassified edge connects the end points of valid classified edges,then it is considered a valid edge link and is not removed from the edge set. This process is done during reconstruction. The resulting unclassified edges have certain properties that can be exploited to weed out spurious edges, which are

- Lower stack id layer edges connecting classified edges are retained
- Higher stack edges must connect to edges with low stack ids near edge endpoints.
- Links or non-classified lines or curves that form t-junctions with classified lines or curves have a higher probability of being spurious edges.

While spurious edges are handled during boundary detection in the next chapter, at this stage an initial candidate set of spurious edges has been identified as links or unclassified edges. These unclassified edges are pruned during boundary reconstruction. Figure 67 shows the unclassified edges in red. Most of these edges are spurious and noisy edges that do not form part of the object form or detail. Green shows lines with direction belonging to D_s while blue shows the line where direction $d_l \neq D_s$. There are only eight possible directions for an edge to follow when a raster is vectorised. Consequently, there is no thresholding required for comparing d_l to D_s, but rather an exact value comparison of the direction, since D_s is a discrete set.

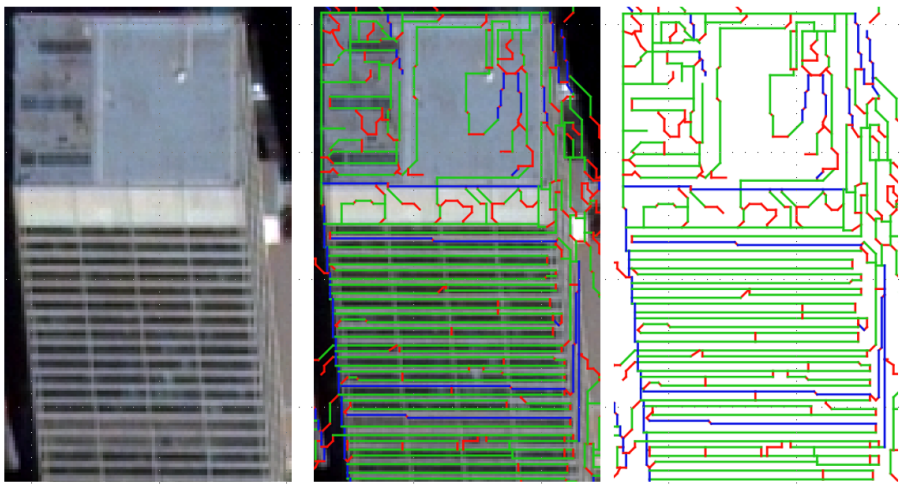


Figure 67: Example Classified edges. The first image is the building used as input to ABED. Second and third images are the classified edges. Blue are lines ($d_l \neq D_s$) with stair stepped effect, green are lines belonging to D_s, and red are candidate spurious edges

5.6 Discussion

A boundary detection and reconstruction strategy has been presented that aims to recover long unbroken lines and curves from satellite images. Most edge detectors aim to recover object boundaries by fitting lines to pixel clusters that fall within a given threshold. From this fitting, the resulting lines are then used as supporting features for other image processing tasks or as estimates for object boundary locations. However, their use in estimating object boundaries is limited due to fragmentation and missing edges. Apart from the limitations presented by satellite images mentioned in Section 2.7.1, applying a local line fitting process with the aim of recovering lines does not always yield correct object boundaries. This is because firstly, object boundaries are assumed to be strictly piecewise-linear. This introduces fragmentation in the case of smooth curves. Secondly, a rotational error is introduced when fitting lines that have small spike errors or corner points that fall within the line fitting threshold as described in Section 5.3.5. To counter this, the line recovery problem is broken down into five different steps. To begin with, the edgels are merged into line segments based on four direction, D_s. This allows us to extract the first set of lines without introducing any rotational errors. Edge connections are maintained, but separated by stair stepping. Secondly, based on stair stepped effect introduced by line to image projection, the rest of the lines are detected. Thirdly, curves are detected, thereby avoiding the problem of over-fitting lines to nonlinear curves. In step four, the remaining edges are classified into four different classes from G_{ls}. By decomposing the boundary reconstruction problem into separate steps, fragmentation is greatly reduced while maintaining accurate edge orientation and connectivity. This decomposition simplifies the detection of curves based on the definition given in Section 5.3.1. Line fitting refinement is attempted at the end in step five to refine line orientations. This result has a positive impact on edge-matching algorithms. However, few spurious edges still remain in G_{ls}. More results are presented in Chapter 7. Efficient boundary recovery paves the way for improved computer vision tasks that rely on edges as features. Two of such tasks are edge matching and 3D reconstruction. From the results obtained from the boundary reconstruction, the next task is to recover the building boundary shapes and perform multiview reconstruction using dense edge features. This is detailed in the next chapter.

6 Building Boundary Shape Reconstruction

Line or edge features have been used largely as support features to reduce matching ambiguities, aid image registration, and reconstruct linear features, such as roads and rivers [236, 184, 67, 80, 48, 237, 237, 238]. Edge fragmentation and missed edges reduce the number of lines that are successfully detected in satellite images. Consequently, the reduced line density limits the use of edges in 3D reconstruction. Chapters 4 and 5 presented frameworks for detecting dense edges and object boundaries in the form of lines while reducing fragmentation and missing edges. However, the lines and curves recovered must be connected into meaningful boundaries that follow object form and detail with high fidelity. The objects of focus in this research are buildings. Satellite images have limited resolution, which hinders the successful recovery of short edges, usually less than four pixels, and corners. These small edges are generally not accurately recovered. This contributes to fragmentation. The building boundary shape recovery algorithm presented here aims to recover object boundaries even when small edges are missing or have not been fully recovered. This chapter forms the second part of the building boundary shape reconstruction framework (ABEDL).

The major advantage presented by the line and curve recovery method in the previous chapter is the ability to represent the edge map as a network of connected lines through a graph. State-of-the-art line recovery methods retrieve lines as independent entities without their interrelationships with adjacent lines [4, 3, 6, 1]. Thus, the line recovery methods only operate locally. This results in uncertain line end points where the lines are supposed to intersect. However, since the edge map from the accumulator preserves connectivity, the algorithm presented here has the ability to reconstruct line intersection points correctly, and the problem of finding closed boundaries can be solved by graph traversal. Thus, a breath-first search algorithm is developed to connect lines and curves into meaningful boundaries. To mitigate the effects of corners and junction that are not detected correctly as a result of pixel mixing, parameter relaxation during traversal is employed to establish shortest path connectivity to nearest valid boundaries using classified links. These boundaries are then matched using some existing line matching algorithms and ground truth to render three-dimensional models. The desirable feature of using dense edge features is that object geometric information is preserved during

reconstruction while avoiding dense pixel matching.

6.1 Building Shape Reconstruction Workflow

The previous chapter presented a boundary detection and reconstruction strategy that retrieves lines and curves from a dense edge map. These recovered lines and curves are used as input to building boundary shape reconstruction. The workflow to reconstruct building boundary shapes has four main steps illustrated in Figure 68.

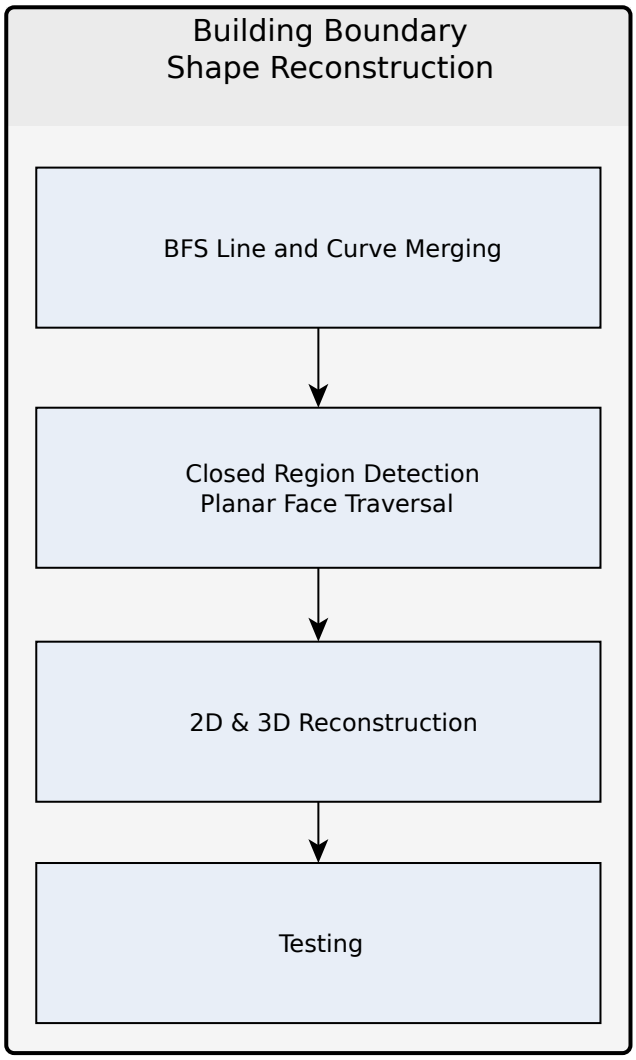


Figure 68: Building Boundary Shape Reconstruction workflow. The workflow is comprised of four steps.

The four steps are:

1. Line and Curve merging: The lines and curves detected in Chapter 5 are connected

using a BFS search algorithm. Parameter relaxation to connect line end points using links is employed. Furthermore, gestalt principles are used to close gaps and connect disjoint lines.

2. Closed-Region Detection: Closed regions are detected using planar face traversal on the line graph.

3. Edge Matching: Lines are matched in multi-view scenes.

4. 3D Model Rendering: The matched objects are rendered in 3D for visualisation.

6.1.1 Line and Curve merging

Chapter 4 demonstrated the framework for dense edge detection. The advantage of dense edge retrieval is that more complete boundaries are recovered, which simplifies the task of boundary reconstruction. Most recovered edges are true positives. In addition to improved straight line retrieval, non-linear curves are detected thereby extending the possibility of retrieving non-parametric shapes. The algorithm used is a Breath-First Search algorithm that searches for connected line entities that form closed boundaries. From Chapters 4 and 5, the edge map is a graph data structure with all edge connectivity information. Lines and curves recovered are stored in the graph $G_{ls} = G(V_{ps}, E_{ls})$ from chapter 5. G_{ls} consist of line segments that have been fit to classified edges. Even though these lines exist on object boundaries, they exist as separate entities that require grouping into boundaries. Before running the BFS algorithm, the graph is further classified into two separate classes, namely line and curve segments from the anchor layer and lines and curves from all other accumulator layers. The anchor layer is the first accumulator layer that contains the least amount of noise as a result of high thresholds and has Identification Number (id) of zero. In G_{ls}, curves have been decomposed into a set of connected piecewise linear segments. The id of the accumulator layer from which a line was derived is stored in the graph. Any other line that extends a line created from the anchor layer with id of zero by being perpendicular or collinear is merged with the line from the anchor layer, and its accumulator level id is updated to zero. This automatically grows the lines. The classification can be expressed as follows:

$$l_{cn} = \begin{cases} 0 & \text{if } ac \equiv 0 \text{ or } l_{ac>0} \cong l_0 \text{ or } l_{ac>0} \parallel l_0 \text{ or } l_{ac>0} \perp l_0 \\ 1 & \text{Otherwise} \end{cases} \tag{6.1.1}$$

where $cn = 0, 1$ is the line class

$ac =$ accumulator level

After classification, the lines are sorted in order of decreasing length in the graph to control the order of traversal. A BFS function is defined as follows.

$$F_{bfs} = F(G_{ls}, f(t, \theta, l_n), r_d,) \tag{6.1.2}$$

where $G_{ls} =$ is the line network graph

$f(t, \theta, l_n) =$ is the line fitting function defined by 5.3.5

$r_d =$ is the parameter relaxation depth

$\varepsilon_{ln} =$ is the line fitting error threshold.

Function F_{bfs} begins with the longest edge at the top of the sorted stack in the graph and selects a path to traverse by analysing the immediate neighbouring lines. Traversal preference is given to the longest edge with the least or equivalent accumulator level to the currently selected one. A further constraint is placed when selecting the path to traverse which is that if two lines are candidates selected for traversal, the one with the least deviation or incident angle is selected and. The function then visits the selected edge and performs two operations. Firstly, check if the combination of the selected line and the starting line forms a longer line within a threshold ε. If a longer line can be fit, the two lines are merged, and the next candidate is selected based on the same criteria. This first traversal extends lines and refines line orientation. If the merging of the lines does not form a new line, the lines are kept separate and search for the next line that can be merged continues with the next line on the sorted stack.

The second function performed by F_{bfs} is to preserve smooth curves that can potentially be incorrectly merged into straight line segments. This is achieved by computing the orientation of line points before a fitting is attempted. The function checks if all the points to be merged by fitting a line are oriented on one side of the line

formed by connecting extreme line points of the lines to be merged. If the line points lie on one side, the lines are not merged and considered a smooth changing curve and the classification updated. Any other lines are ignored during traversal if they are less than or equal to three pixels in length and do not qualify for merging. Lines greater than three pixels with and id of zero are marked as potential boundaries. This algorithm is shown in algorithm 2.

Algorithm 2 Breadth First Search boundary recovery algorithm

Require: $G_{ls} \neq NULL$
Require: $f(t, \theta, l_n)$ ▷ Input: Line fitting function
Require: $r_d = 5$
Require: $P_l \neq NULL$ ▷ Points on a given line
Require: $\varepsilon_{l_n} = E_(\varepsilon)(l)$
Require: sort(l_n) in G_{ls} ▷ By decreasing length and ac level
  while l_i is not visited **do**
    Set $l = visited$
    for each out-neighbor l_i of l_n **do**
      Set $P_l = P(l \ \& \ l_i)$
      if $f(t, \theta, P_l) \leq \varepsilon_{l_n}$ **then**
        Set $l = l + l_i$
        Set $l, l_i = visited$
        Set $P_l = P(l + l_i)$
      end if
    end for
  end while

The results of this first process are longer lines with improved orientation. However, connectivity of these boundaries must be established. This is achieved through parameter relaxation. An example of recovered curve through BFS search algorithm is shown in Figure 69. Compared to other line detectors, ABEDL recovered most lines as shown in image (f) on the bottom right. As previously mentioned, short boundary segments are not recovered with high accuracy. The longer lines are connected by small segments or links that are at times jagged. However, the connected curved boundary is recovered with interior detail in comparison to the other detectors.

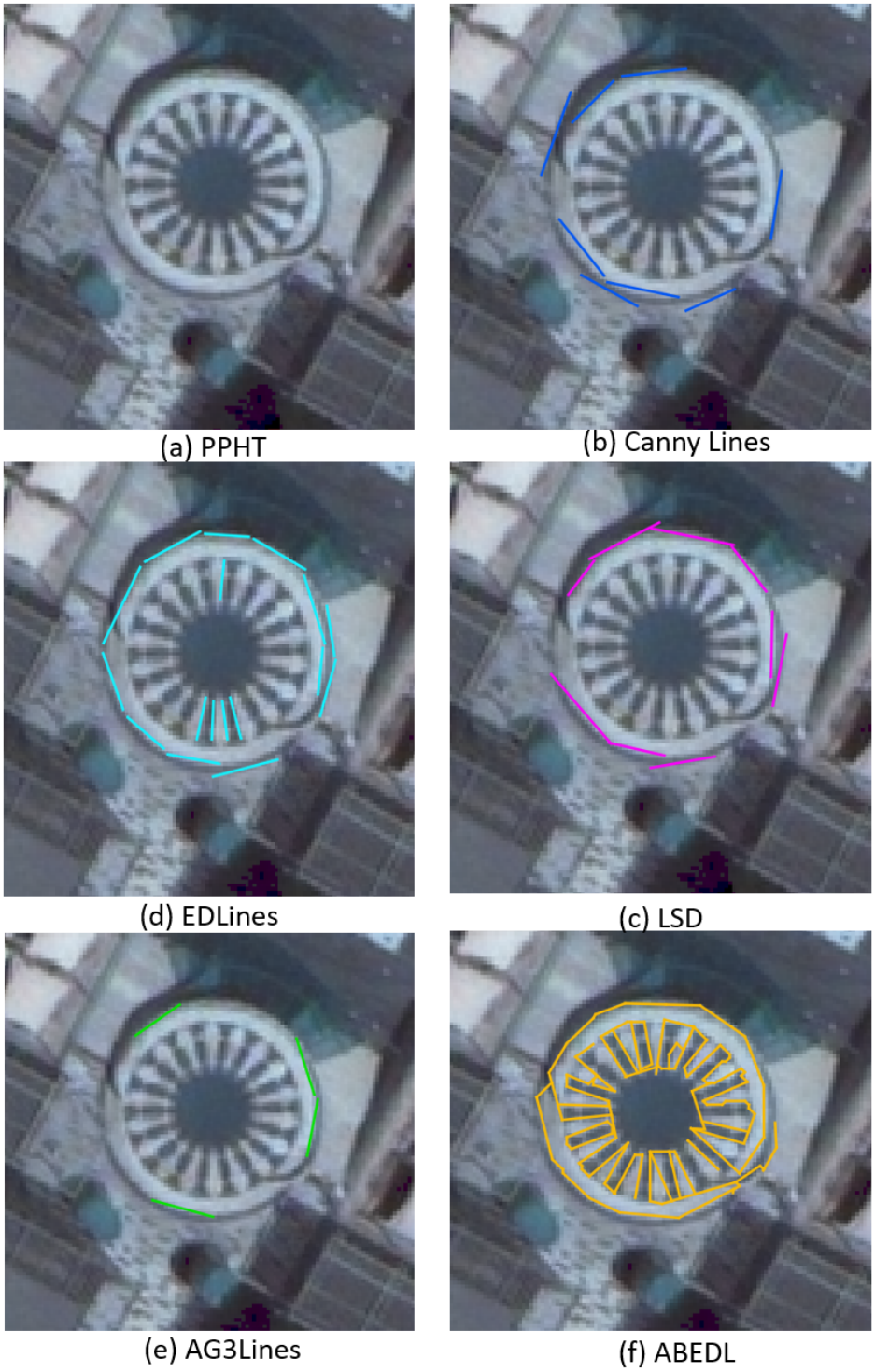


Figure 69: Image (f) from ABEDL shows the recovered boundaries of a circular building structure. The curved nature of the boundary is retained. The recovered boundaries are connected, with interior detail retained. ABEDL shows an improvement over the detail recovered by PPHT, CannyLines, EDLines, LSD and AG3Lines. Most of the linear segments were recovered. This level detail benefits wireframe reconstruction. The other five methods have a high number of fragmented and missing boundary segments. With ABEDL, the short boundaries are recovered, but with limited accuracy, since the number of points on these short edges is insufficient to fit a line.

6.1.2 Merging by Parameter Relaxation

The above algorithm merges and fits longer boundaries on condition that an immediate collinear neighbour exists. The second task is to traverse the graph and find edge links that are not collinear or perpendicular, but rather connect valid edges at end points. Lines that have been detected are connected by links or edges classified as links. Parameter relaxation allows the shortest path to be traversed for the next valid boundary to be discovered. The shortest paths are referred to as boundary links. To connect the nearest valid lines, parameter relaxation is applied to the Function F_{bfs}.

By relaxing all parameters for a depth r_d, all lines not marked as boundaries are used to find the shortest path to the next valid adjacent boundary lines. Most of the lines traversed in parameter relaxation have been classified as links in the edge classification process presented in Chapter 5. The shortest path traversed to the next valid boundary line from a valid boundary line is recorded. The next valid edge is then compared with the last valid edge visited based on gestalt principles of continuity. If the lines connect in a meaning-full way, the shorted path is considered a valid edge connection between the lines and is classified as a valid boundary link. In this way, gaps between boundaries are filled where a path exists. The extension of algorithm 2 with parameter relaxation, where traversal is allowed without any attempt to fit a line during line search shown in algorithm 3. A depth r_d, is used to control the depth of the search path. The traversal is constrained to follow edge links. If a valid line is found, the lines are connected based on gestalt principles of closure, continuity, and proximity. However, a link will not always exist because of missing edge pixels or occlusions. A solution to missing links is proposed in Section 6.1.4.

6.1.3 Validating Links

After detecting lines and curves, the edges that remain unclassified and do not form valid boundary links are still connected to the classified lines and curves.The first case of edge linking occurs when a reconstructed valid line or curve shares a second degree vertex with a path traversed during parameter relaxation. Linking with unclassified edges begins by classifying all traversed shortest-path links that share an end point vertex with a valid

Algorithm 3 Breadth First Search boundary recovery algorithm with parameter relaxation

Require: $G_{ls} \neq NULL$
Require: $f(t, \theta, l_n)$ Input: Line fitting function
Require: $r_d = 5$
Require: $P_l \neq NULL$
Require: $\varepsilon_{l_n} = E_(\varepsilon)(l)$
Require: sort(l_n) in G_{ls} by decreasing length and ac level
  while l_i is not visited **do**
    $l_i \leftarrow visited$
    for each out-neighbor l_i of l_n **do**
      $P_l \leftarrow P(l \ \& \ l_i)$
      if $f(t, \theta, P_l) \leq \varepsilon_{l_n}$ **then**
        $l \leftarrow l + l_i$
        $l, l_i \leftarrow visited$
        $P_l \leftarrow P(l + l_i)$
      end if
    end for
  end while
  Set $r = 0$
  while l_{uc} is classified as link and out-neighbor is classified **do**
    $l_{uc} \leftarrow visited$
    for each out-neighbor not classified l_{uc} of l_n **do**
      if then$r \leq r_d$
        if l_i is unclassified **then**
          $l_{uc} \leftarrow l_c + l_i$
        else if l_i is classified **then**
          $l_{uc}, l_i \leftarrow$ visited and classified
          break traversal (Break)
        end if
        $rd \leftarrow rd + 1$
      end if
    end for
  end while

line or curve. If any of the traversed links share one or more end-point vertices with two or more valid lines or curves, the link is retained as a valid boundary link. If multiple paths link to the same lines or curves, the shortest path is selected. By linking these shortest paths with valid lines and curves, the boundaries are connected, extended, and closed. Figures 70, 71, 72 show examples of boundary closures using edge links. Yellow represents the shortest path links that connect valid edges in the graph G_{ls}.

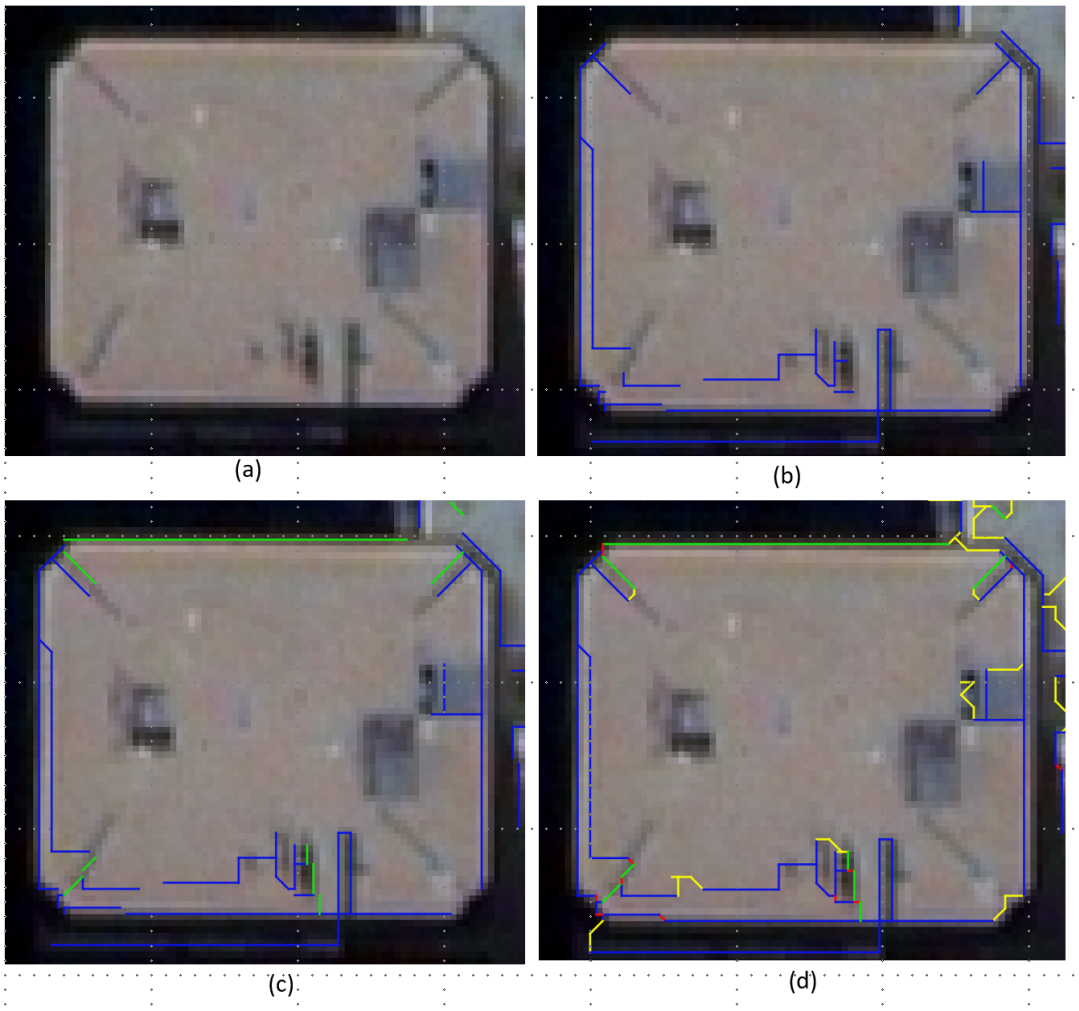


Figure 70: Figure illustrating boundary merging by edge linking. Image (a) show the input image. Image (b) shows the first set of lines (blue) that are valid as described in section 6.1.1. Image (c) is the second set of lines that do not belong to D_s but are valid. Image (d) shows linking of the edges using unclassified edges. The link edges are shown in red and yellow. By linking edges, the boundary has been closed.

6.1.4 Merging using Gestalt Principles

Edge links are not always guaranteed to exist. Missing links result in dangling line and curve end points. To complete these boundaries, gestalt principles are used, mainly,

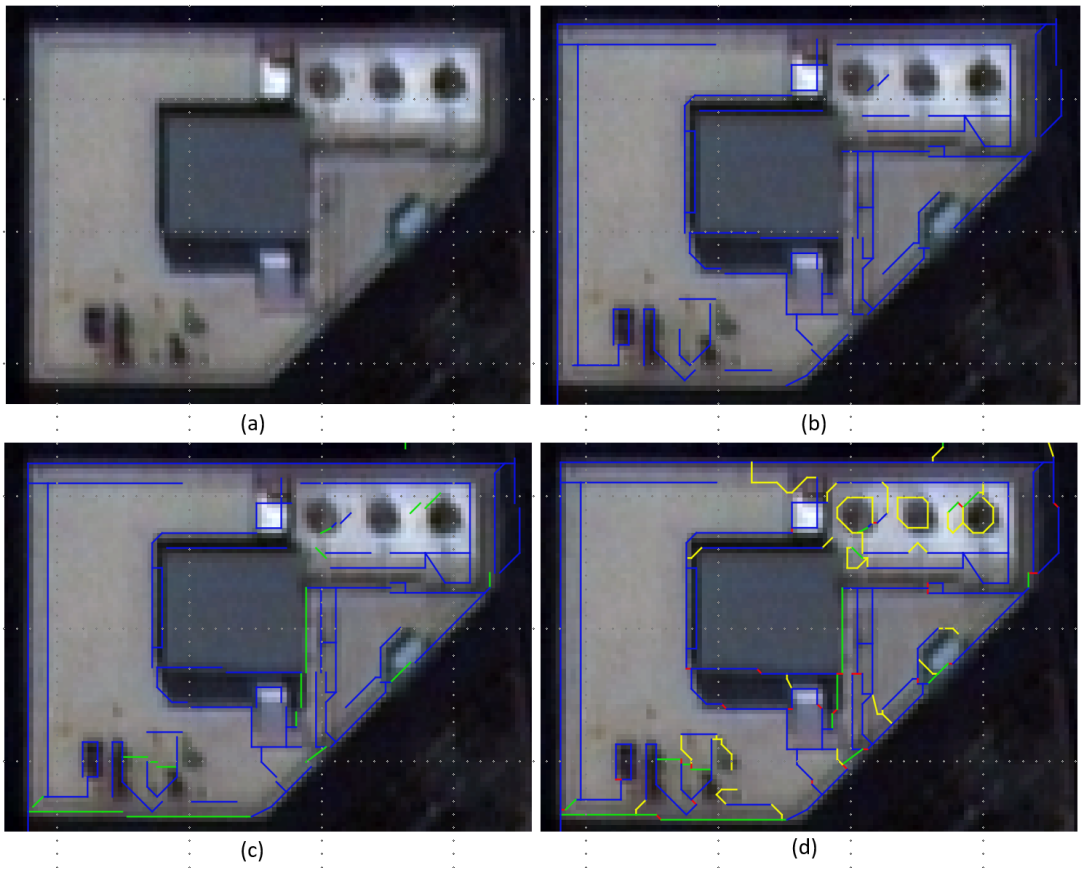


Figure 71: Figure illustrating boundary merging by edge linking when there is an internal superstructure. Image (a) show the input image. Image (b) shows the first set of lines (blue) that are valid as described in section 6.1.1. Image (c) is the second set of lines that do not belong to D_s. Image (d) shows linking of the edges using unclassified edges. The link edges are shown in red and yellow. By linking edges, the boundaries of some small structures have been closed.

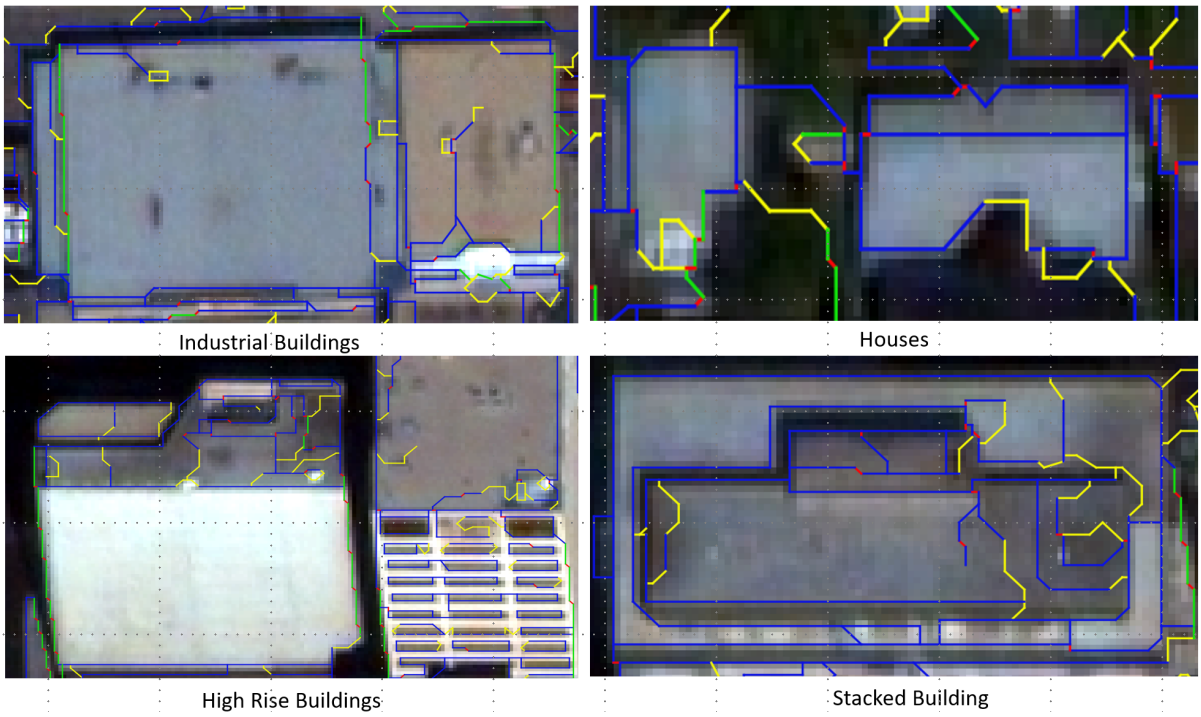


Figure 72: Different building objects with boundaries that have been linked and closed by using unclassified edges. The framework attempts boundary closure for even the small objects. The four images above show different cases of where boundaries were closed using links.

closure, proximity, continuity, and symmetry. Using proximity, non-parallel edges that are close together but disjoint are assumed to belong to the same object boundary. Using symmetry, if a building object is not completely bounded by edges, the first assumption is that the building has a line of symmetry, or the corners can be mirrored along a diagonal. This is used to search for the existence of opposite corners. Beside the proximity rule, other rules are merely used as search guidelines. Since a boundary segment is always guaranteed for all objects, albeit disjoint and incomplete, gestalt principles are used as search space constraints. In summary, using proximity, lines are intersected, and using symmetry and patterns, partial building boundaries are used to define a search space for missing boundaries using mirroring, diagonals, and bounding boxes. The first step to using gestalt principles is to identify patterns in the data, namely parallel lines, closed regions, repeated patterns, and nearest neighbour information for all linear and curved segments. This is achieved by spatially indexing each edge in graph G_{ls} using an R-Tree. From the recovered boundaries in the R-Tree, key-edge arrangements are created based on gestalt principles. It is important to note that the resolution of the satellite image is limited, and thus the ability to detect short edges is limited. A fair amount of estimation

based on gestalt principles will assist in improving boundary completeness.

6.1.4.1 Key edges selection: As highlighted in [74, 126, 239], matching edges in groups or exploiting neighbouring edges provides more information about the structure of the object. However, care must be taken when establishing edge relationships, since the probability of similar repetitive patterns in large scenes is high. The first step is to prune the boundaries into key-boundary sets. Firstly, edge arrangements are selected based on the law of similarity, proximity, closure, and continuity. Firstly, parallel lines are assumed to belong to the same object if their lengths are almost similar within a threshold. Given an image, let the entire line set be $\varepsilon_s^i = \{e_1^i, e_2^i ....e_n^i\}$ where n is the number of valid lines from the image. A key line or boundary set is thus defined as

$$\kappa_a^i = \{e_c^i\}_{c=1}^f : \kappa_a \subset \varepsilon_s^i$$

where f is the total number of lines in the set κ_a^i.

Selecting the key boundary set begins by selecting any given starting edge, e_j^i of length l, with $0 \leq j < n$ and $l > l_\epsilon$. l_ϵ is a length threshold that is reduced by half in an iterative loop. A selected candidate segment has two flanking regions R_r and R_l. The flanking regions are defined by dividing the xy plane and the line created by the selected segment as illustrated in Figure 73. Each flanking region is divided into three regions, namely, start, middle, and end. For each of the sub-regions, adjacent lines that potentially intersect the selected segment in the start and end regions are searched for. The intersection angle θ is chosen to be 75^o, to avoid near parallel lines. Adjacent lines are then placed into the appropriate sub-regions. The start and end sub-regions will have potential candidates for boundary completion. Lines that straddle all three regions are candidates for parallel neighbours. The bounding regions enclosing the candidate segment and the selected adjacent lines are defined on each of the flanking sides. Finally, any other segment that is enclosed or overlaps significantly with the bounding region is selected to be part of the key boundary set, and a group of key boundary features is created.

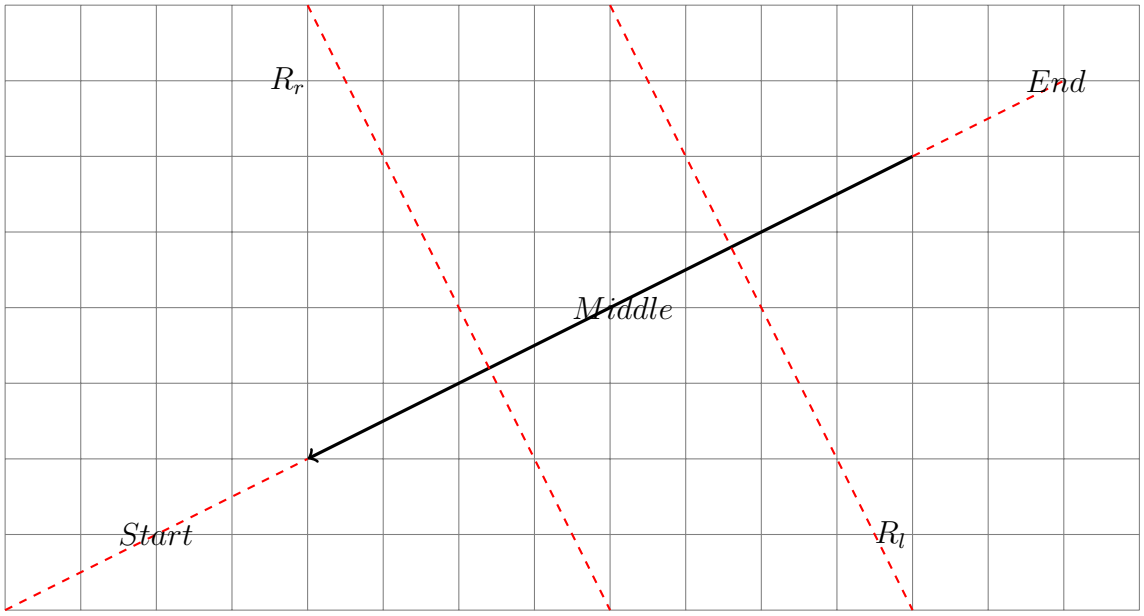


Figure 73: Space partitioning for identifying key edge groups. The space around an edge is partitioned into five regions namely the right region R_r, the left region R_l, the start, end and middle regions. Edges in the start and end regions that are not parallel or are curves are selected to form key edge groupings with the main edge.

6.1.4.2 Key edge group merging: The process of defining bounding regions is repeated for any salient edge that has not been assigned to a group of line features until all possible sub-regions and key boundary feature groups have been identified. Due to the existence of occluding surfaces, stacked man-made structures, or substructures, overlap between key edge feature groups occurs. When overlapping sub-regions overlap, there are two possibilities with regards to the overlap. Either there exists an occluding structure or its a stacked structure or sub-structure. Using intersection, and extension, boundaries are completed where a link does not exist. The topic of boundary estimation using Gestalt is vast and was not fully explored in this research.

6.1.5 Closed Region Detection

After closing and linking boundaries, closed regions need to be identified. This is achieved by using a planar face traversal algorithm on G_{ls} [240]. Planar face traversal decomposes the entire image region into closed polygons. The result from planar face traversal is a set of closed region that includes building boundaries and, in some cases, the interior space between buildings. Closed region detection applies to 2D building boundaries. In the case of nadir images or orthophotos, the recovered boundaries are true object boundaries,

since occlusions are minimal. However, with multiview images, reconstruction is first attempted. Because of projective distortion, the closed regions recovered from off-nadir images are not the true building boundary shapes. However, recovering 3D boundaries yields the true positions and heights of building boundaries. Boundary estimation can then be attempted after reconstruction.

6.1.6 3D Reconstruction and Rendering

In the case of orthophotos the resulting boundaries can be considered as the final output in 2D. These results can be used directly for downstream applications. However, most satellite images are acquired with an off-nadir angle, the recovered boundaries have projective distortion and will not be useful unless the object boundaries are recovered in 3D. The matching of boundaries is, however, outside the scope of this research. Consequently, the recovered edges are matched using existing line matching algorithms and ground truth to recover the 3D boundaries as a detailed wireframe. Line-matching algorithm presented by [125] is used. The method presented by [125] uses junction points, where lines intersect, to generate line descriptors that are invariant to rotation and scale through the Scale Invariant Feature Transform (SIFT) [129]. The interior junction point descriptors are considered more robust due to defined locality and are less affected by neighbouring structures since they are located in the interior of an object. The results that are presented are based on the results of this line matching strategy. Since the performance of the line matchers has not been evaluated on satellite imagery, ground-truth based matching is used to augment the final outputs. However, it is important to note that most line matching algorithms are designed to match linear segments exclusively. Consequently, failures in matching links and curves were encountered.

To reconstruct the wireframe model, the matched segments are first used to compute the line positions in 3D. The grouping of key edges described in Section 6.1.4 is maintained in 3D. Each 3D line is assumed to exist on a plane that intersects with the ground from the line. Recreating these planes around a key edge group creates planar areas that are then projected back to image space using the given RPCs. Occluded facets can be identified using RPCs. Using the direction of the normals of facets, which for occluded

facets is the opposite of the viewing direction, occluded facets are identified and removed. After reprojection, the area covered by each facet in image space is augmented with any unmatched lines and curves that are in the interior of the facet. This process adds the wireframe detail. The facet together with the augmented detail, is projected back to 3D using RPCs. This gives the final wireframe model.

If edge maps from more than two views are produced, then edge matching is performed in image pairs. if more than two image views exist, each image is paired with a different image, and matching is performed across all image pairs. For all correct edge matches, the 3D positions of the matched edges coincide on the ground. By comparing the edge intersection points in 3D, matched edges are verified. Put differently, having more than two views introduces an independent matching verification method. If salient edges do not have matches across scenes, potential matches are found by back-projecting 3D edges computed from other views onto the scenes that have edges without matches. By comparing the back projected edges with the edges of the scene without matches, new edge matches are established and augmented to the 3D model. This is considered incremental reconstruction. Figure 74 shows an example wireframe model for Calgary, site TI_3.

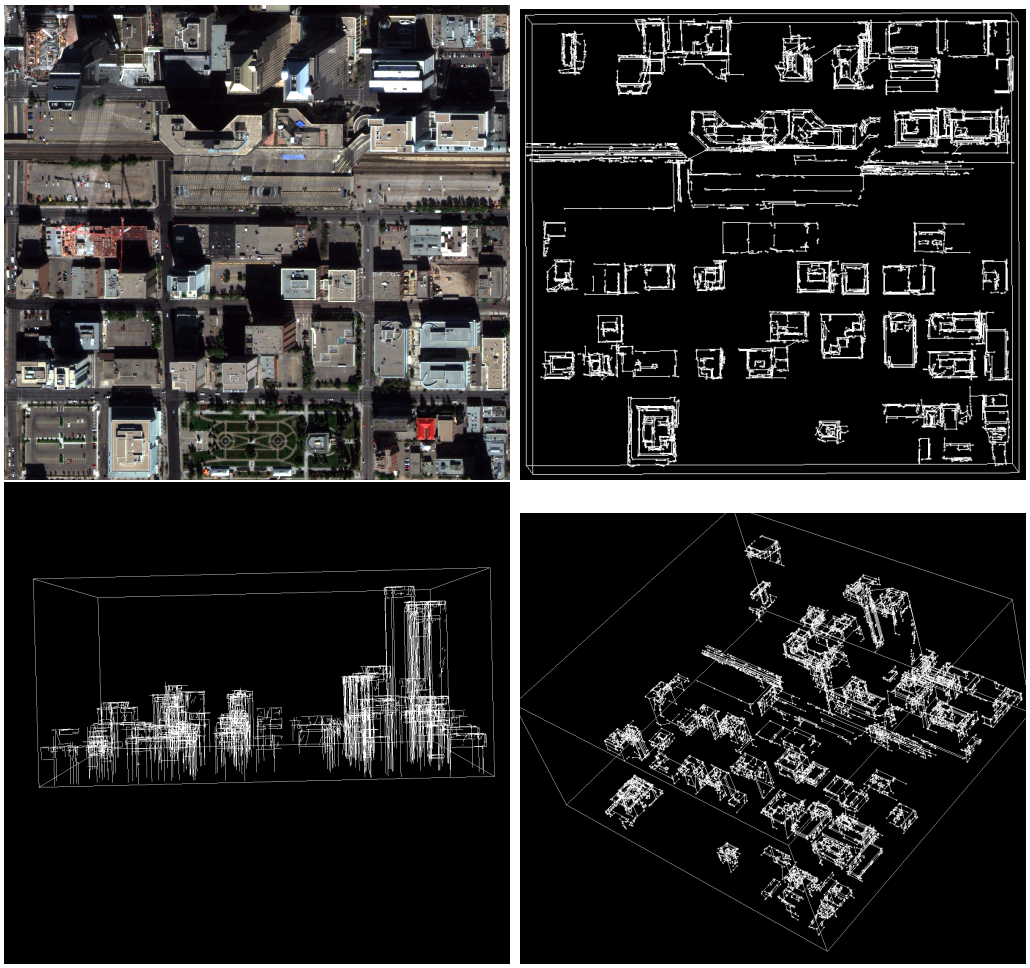


Figure 74: 3D Wireframe model of Calgary created from reconstructed boundaries by edge matching. The top left image is on of the input image, top right shows the top view of the wireframe. The superstructures and parapet boundaries are visible. Bottom left shows the side view with some facade details reconstructed.

7 Results

This chapter presents qualitative and quantitative assessments of the results from ABED and ABEDL frameworks. The first section of this chapter gives a description of the data used in the research. The next section presents the results of ABED from the different test sites selected within the described datasets. Following the results from ABED, ABEDL results are presented and evaluated in Section 7.3. Section 7.4 presents some results of the attainable 2D and 3D boundary reconstruction. Finally, performance evaluation results on the framework are presented followed by brief conclusions. Chapter 8 gives a detailed analysis of the results presented and conclusions drawn in Chapter 9.

7.1 Data Description

Five datasets from different geographical regions of the world are used for the analysis of the results. Four of the datasets have stereo or multiview images. The first dataset is of Calgary in Canada. The dataset was acquired by the WorldView 2 satellite at $0.5m$ resolution. The dataset has three multiview scenes. The acquisition date of the first two images, which are dedicated stereo images, is 08-10-2012, and the third scene was captured on 24-10-2012. The captured area is approximately (29km2). The geographic region ranges from longitude -114.11619843^o to -114.02712079^o, and latitude 51.02504370^o to 51.06740571^o north. The mean off-nadir viewing angles are between 14^o and 31^o. The second dataset is $0.3m$ resolution imagery of Brisbane, Australia. The dataset consists of three multiview images captured on 15-05-2014, 14-11-2014 and 14-12-2014 respectively. The area covered is approximately $39km^2$. The geographic region covered ranges from longitude 152.99721267^o to 153.05193823^o, and latitude -27.49751759^o to -27.44441406^o. The mean off-nadir viewing angles are between 25^o and 28^o. The third dataset is of London. This dataset contains two images at $0.5m$ resolution. The bounding region is defined by longitude -0.39121321^o to -0.27683239^o and latitude 51.38896598^o to 51.49060608^o. The images were captured on 15-10-2015. The off-nadir viewing angles are 21^o and 26^o. The fourth dataset is of Dubai and contains three images. Two of the images were acquired on 11-06-2018 and the third image on 27-04-2018. The geographic region covered by the images is between longitude

55.20458508^o and 55.34441106^o and latitude 25.12175265^o and 25.24864119^o. The images were captured at 0.3m resolution with off nadir viewing angles of 23.6^o, 24.8^o and 27.1^o respectively. The fifth dataset is a single image of Mourmelon, France, captured at 0.5m resolution and covering a geographic region of longitude 4.354078^o to 4.376986^o and between latitude 49.134557^o and 49.143010^o. All images are true color composites and have been corrected for image biases described in Section 2.2.2.2. Figure 75 shows two images of Calgary and Brisbane, respectively.

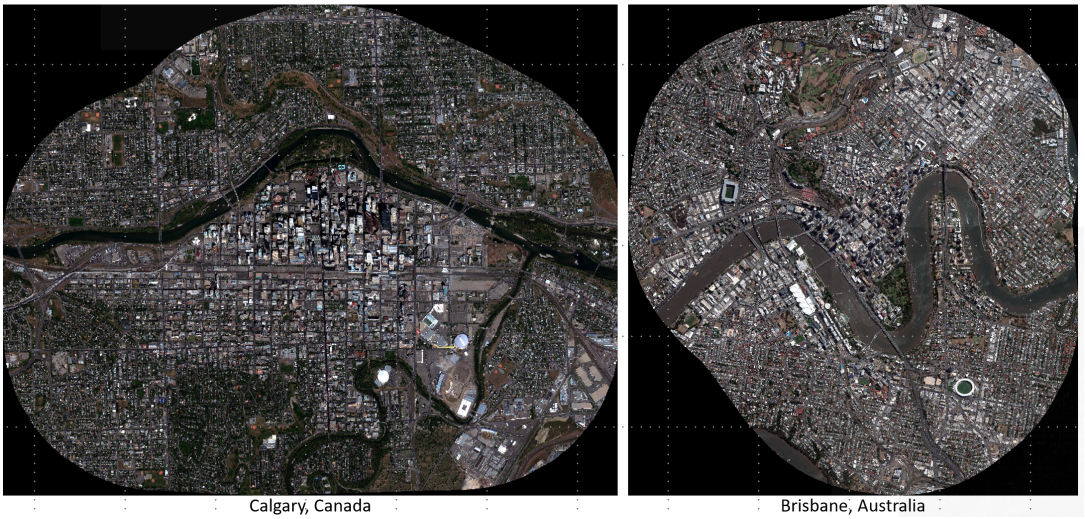


Figure 75: Left image shows Calgary, in Canada and the right image shows Brisbane, Australia. The images were captured at 0.5m and 0.3m respectively.

From the different regions covered by the datasets, ten different sub-images were selected to create ground truth images. The outer boundaries of the building and the features of the facade were manually labelled, producing detailed object form and detail. The ten sites were selected in areas that reflect different architectural styles. Three regions were selected from Calgary, three from Brisbane, two from London, one from Dubai, and one from Mourmelon, France. Table 2 below shows the different images selected, the region from which they were selected, and the size of the image.

7.2 ABED Results

The aim of developing the ABED framework was to improve edge detection by retrieving longer unbroken edges, which are subsequently used to reconstruct building object boundaries. The next section presents a qualitative and quantitative assessment of the results obtained by ABED. Firstly, edge detection results of ABED are shown for

Test Images Selected For Evaluation				
Image	*City*	*Dimensions*	*Resolution*	*Type*
TI_1	Brisbane	1441 x 924	$0.3m$	CBD, Low Rise Buildings
TI_2	Calgary	1184 x 1110	$0.5m$	CBD, Low Rise Buildings
TI_3	Calgary	658 x 623	$0.5m$	CBD, High Rise Buildings
TI_4	Brisbane	1942 x 1688	$0.3m$	CBD, Industrial
TI_5	Dubai	1913 x 1770	$0.3m$	CBD
TI_6	Brisbane	917 x 679	$0.3m$	Industrial, Low Rise Buildings
TI_7	Calgary	400 x 413	$0.5m$	Mall, Commercial Buildings
TI_8	London	503 x 376	$0.5m$	Residential
TI_9	Mourmelon	1001 x 656	$0.5m$	CBD and Residential
TI_10	London	1107 x 1018	$0.5m$	Residential

Table 2: Ten test image sites selected for evaluation from different zones with varying size and resolution.

the different sites and compared with two state-of-the-art methods. For all ABED results, 153 accumulator levels were traversed to generate edge maps.

7.2.1 ABED results

The first results show a visual comparison of ABED with two state-of-the-art algorithms, namely CannyPF and EDPF. These two algorithms were selected because they implement automated parameter tuning when detecting edges. Canny edge detector was not included since CannyPF is an improved Canny Edge detector with automated parameter tuning. Figures 76, 77, 78 and 79 show the edge detection results on then ten sites for CannyPF, EDPF and ABED. The results of ABED show a denser edge map with object form and detail. While CannyPF and EDPF appear sparse and cleaner, a lot of detail of the objects is missing. To illustrate the missing detail, Table 80 shows a zoomed comparison of some individual building objects selected from the ten sites. From Figures 80 and 84, the difference in detail is evident between the detectors, with ABED providing more detail. More detail benefits reconstruction of building boundaries. However, as shown in the last sample in Figure 79, the presence of vegetation increases the density of the edges since the edges of the trees are also recovered.

Figure 80 illustrates the level of detail recovered by ABED on specific building structures selected from the sites. This illustrates that the dense edges of 76 are relevant edges that show the detail of the objects. The localisation filter ensured that

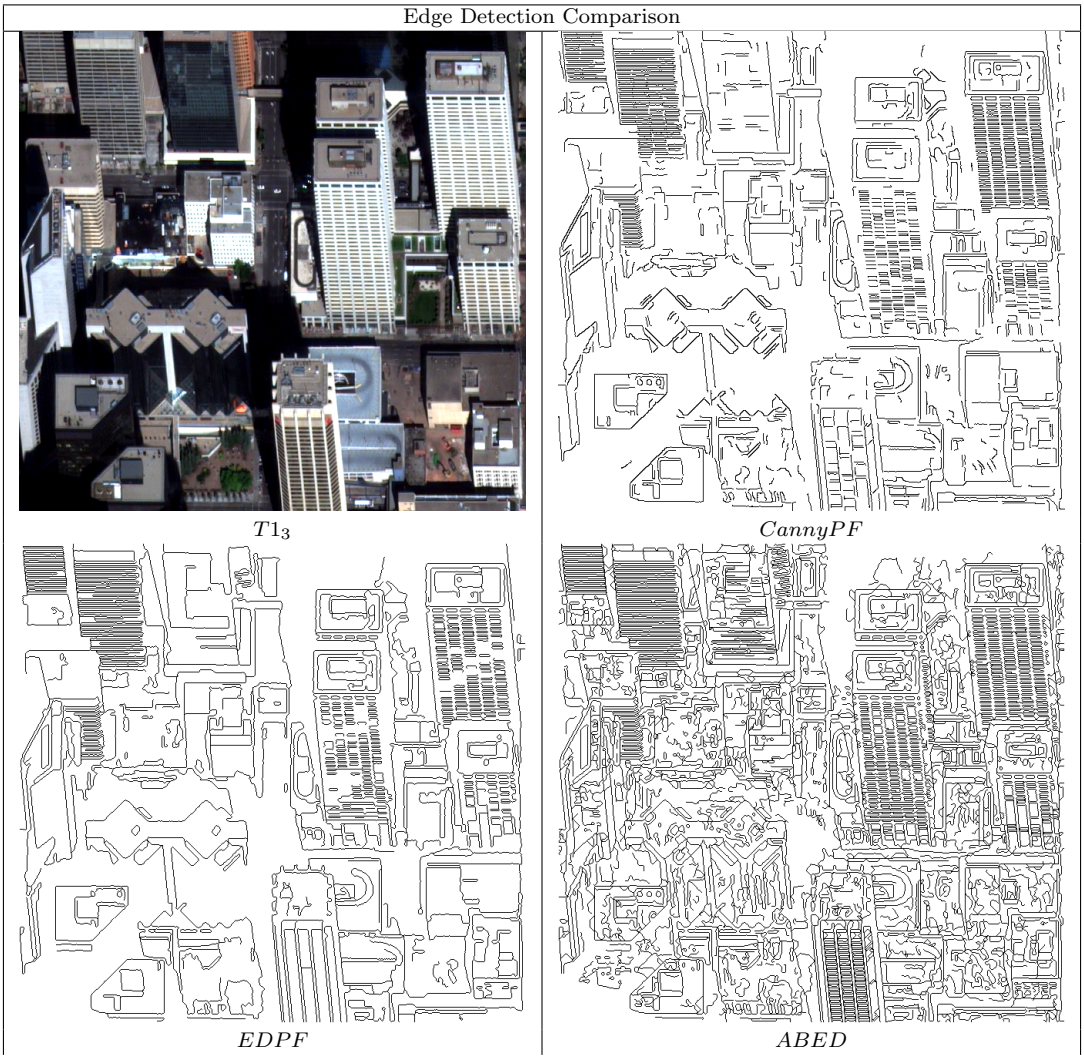


Figure 76: Edge detection results comparison of ABED versusCannyPF and EDPF for Calgary, image TI_3. ABEDL, has denser edges in the bottom right image in comparison to the other detectors in a CBD area.

Figure 77: Edge detection results comparison of ABED versusCannyPF and EDPF for Brisbane, image TI_7. ABEDL, has denser edges in the bottom right image in comparison to the other detectors in a CBD area with reduced fragmentation

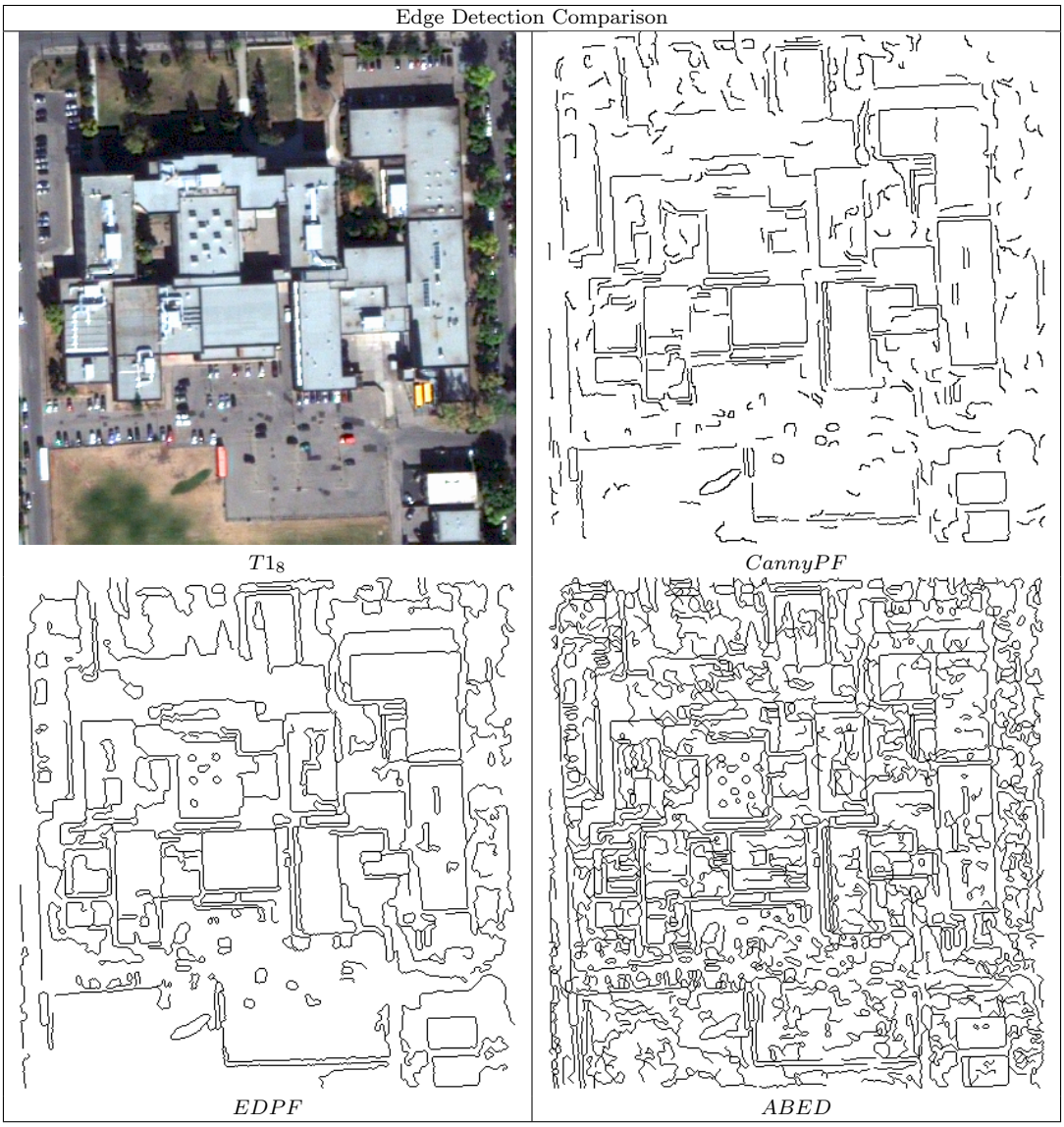


Figure 78: Edge detection results comparison of ABED versusCannyPF and EDPF for Calgary, image TI_8. ABEDL, has denser edges in the bottom right image in comparison to the other detectors around a large structure with roof detail recovered. The other two methods have missing edges for object detail as shown by the sparse edges in EDPF and CannyPF.

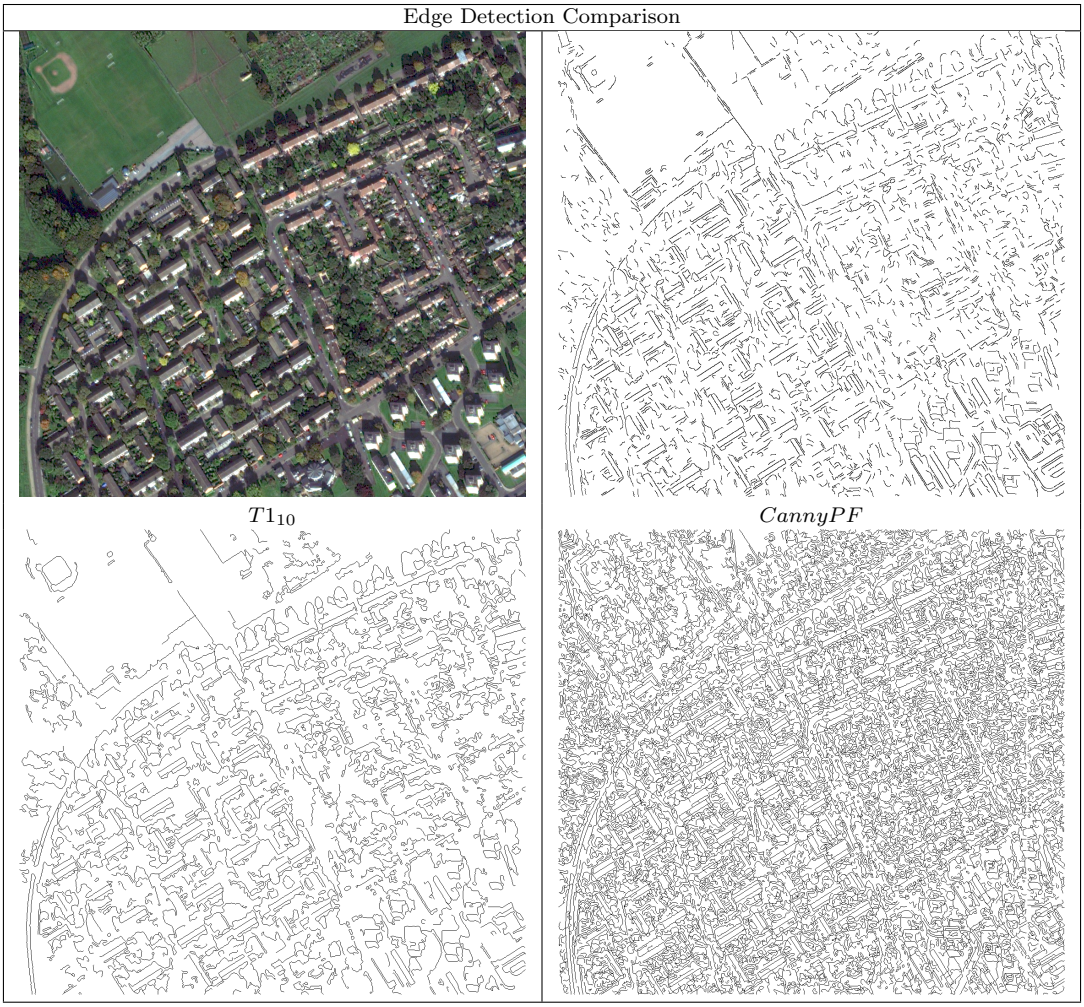


Figure 79: Edge detection results comparison of ABED versusCannyPF and EDPF for sample T_{10} appears noisy but the edges are relevant tree edges or short boundary segments in a residential area with small objects. However, these dense edges benefit the boundary recovery process.

Figure 80: Edge detection results comparison of ABED versusCannyPF and EDPF for the ten sites, TI_1 to TI_{10} on specific buildings. ABED results shows relevant edges on the buildings facades that were not detected by CannyPF and EDPF. In all cases, the details of the facade and roof structures are visible with ABED.

relevant edge pixels were kept using the edge validation function.

Figure 81 shows the extra edges that ABED contributed when compared to Sobel edge detector on a image of Brisbane with a resolution of $0.3m$. Yellow edges are produced using Sobel edge detection and blue lines are the extra edges recovered using ABED . Green circles highlight important edge detail that was detected by ABED. In some cases, where colour variations exist, edges are recovered as shown by the red circle and where vegetation exists. The goal of ABED is to recover such object detail.

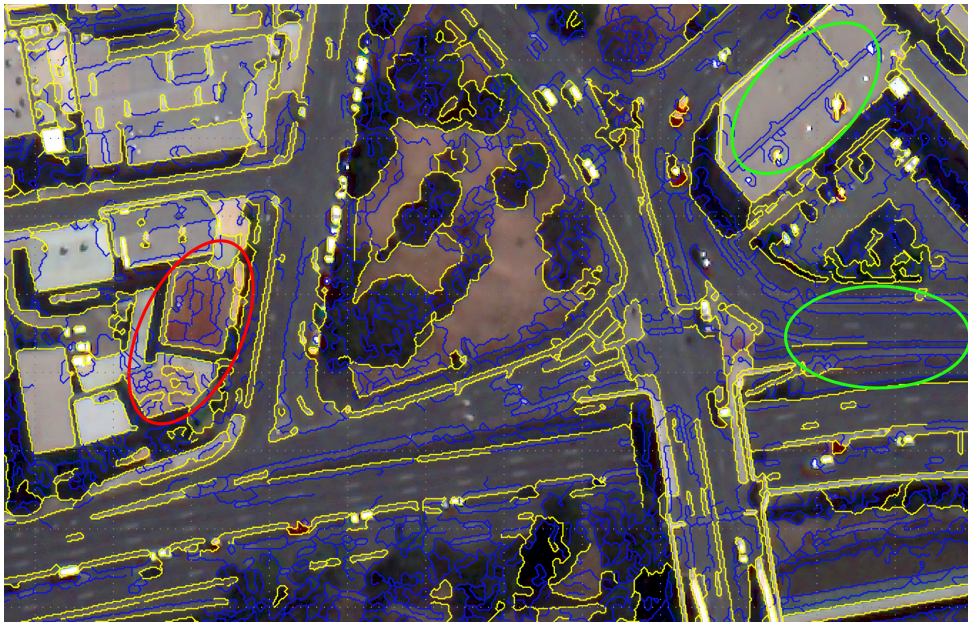


Figure 81: Comparison of ABED and Sobel edge detector. Yellow edges are initially detected using the Sobel operator. Blue shows the edges that were detected by ABED. ABED recovered more edges when compared to the Sobel operator. This is because Sobel uses a single threshold while ABED is parameter free. In most cases, ABED grows edges to form longer unbroken edges. Slight color variations on building roofs are detected as shown by the red circle. This shows the performance of ABED over parameter based edge detectors.

Figure 82 shows Canny edges in yellow, retrieved with a high and low threshold of 200 and 30 overlaid with the results of ABED in a Central Business District in Brisbane. Canny edge detector detects fewer edges compared to ABED. Canny edge detector, misses weak edges while avoiding noise, which is in fact relevant object detail. However, ABED (blue edges) is able to improve on the results capturing most of the missed boundaries and interior object detail. The green ellipses show some of the dark colored facades where edges were detected by ABED.

Figure 83 shows a comparison of the Canny edge detector with a high and low

threshold of 200 and 30 in yellow and ABED in blue, in a residential area. This is an area in Calgary at $0.5m$ resolution. ABED retrieves road lines, and attempts to extract roof ridge lines in the areas marked with red dots. The reduced resolution of satellite, limits the detail that is captured for small objects, resulting in limited accuracy of the edges recovered. This is evident with the roof ridge lines. However, more relevant boundaries on buildings are recovered in comparison to Canny edges. This is because thresholding results in fragmented and missed edges.

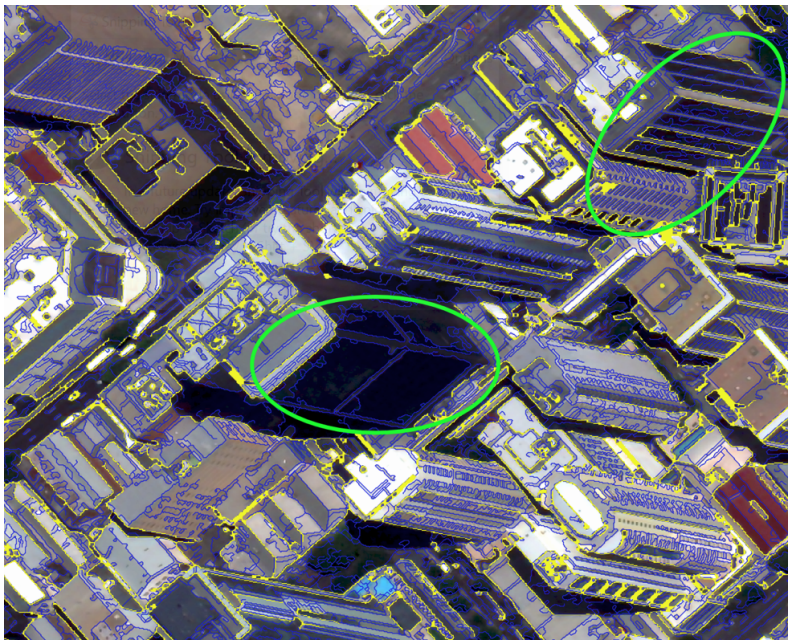


Figure 82: Comparison of ABED (blue) and Canny edge detector (yellow) in a CBD. Yellow edges are detected using the Canny operator. Blue shows the edges detected by ABED. Most edges are recovered by ABED. Canny edge detector missed edges because of thresholding. This shows the advantages of using automated parameter tuning over thresholding. Most object form and detail are recovered by ABED in dark regions, as shown in the circled regions.

The aim of ABED was to improve edge detection for building objects. The aim was to recover all relevant edges under varying imaging conditions with reduced fragmentation. Figure 84 shows the results of ABED compared to CannyPF and EDPF under difficult image conditions. Edges that were not detected in the low-contrast and dark regions by CannyPF and EDPF are recovered by ABED for most buildings.

Finally, to illustrate the advantages of using automated parameter tuning, the edges of the images in Table 85 were classified according to the level of the accumulator that contributed to their detection and subsequent growth of the edges. Green edges are the salient edges that are initially detected at the bottom level of the accumulator. In

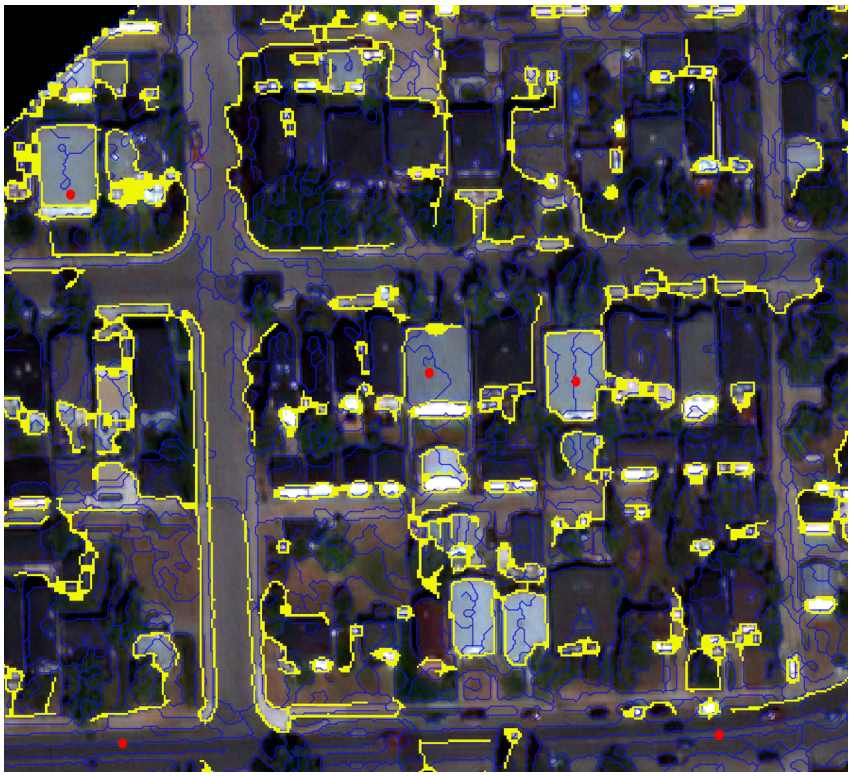


Figure 83: Comparison of ABED (blue) and Canny (yellow) in a residential area. ABED over-detects small objects such as trees. However, ABED partially detects gable roof ridge lines that were not detected by Canny edge detection, shown by the red dots, ABED recovers most of the missed roads and captures boundaries missed by Canny on some houses. This level of detail that is missed by canny results in incomplete building form, while ABED detects edges representing form and detail. This shows the advantages of automated parameter tuning over thresholding.

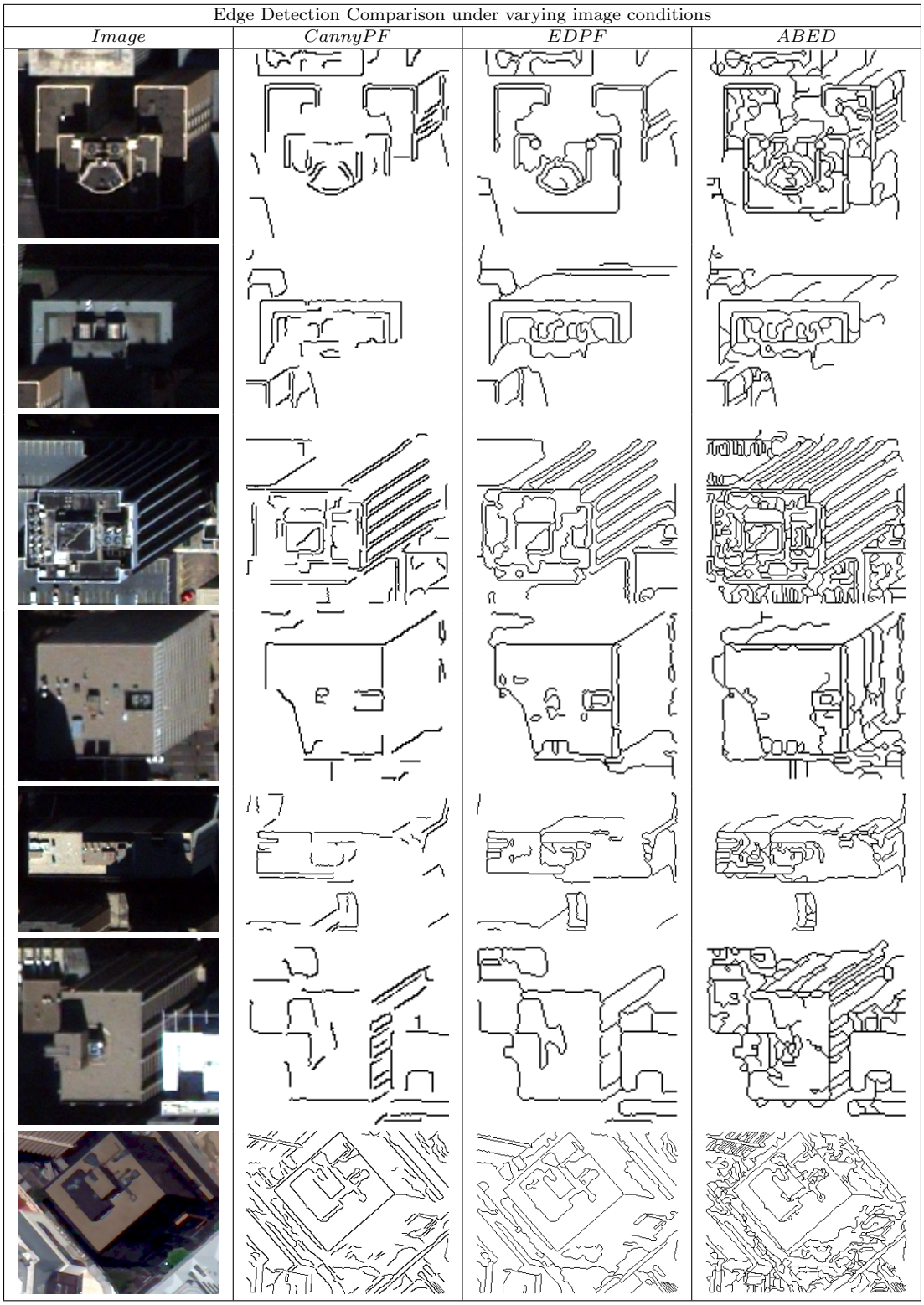


Figure 84: Edge detection results comparison under varying conditions. The images in the first columns have shadows and dark regions around them. The results from CannyPF and EDPF in the second and third column miss building edges as a result of the varying imaging conditions. ABED in the last column recovered most of the edges of the buildings in the dark and shadowy regions in all cases.

addition to the green colour, each colour represents the level of the accumulator where that edge was detected. A significant number of edges were detected at other levels, while maintaining good localisation on building structures. This was the main goal of the first objective achieved with ABED.

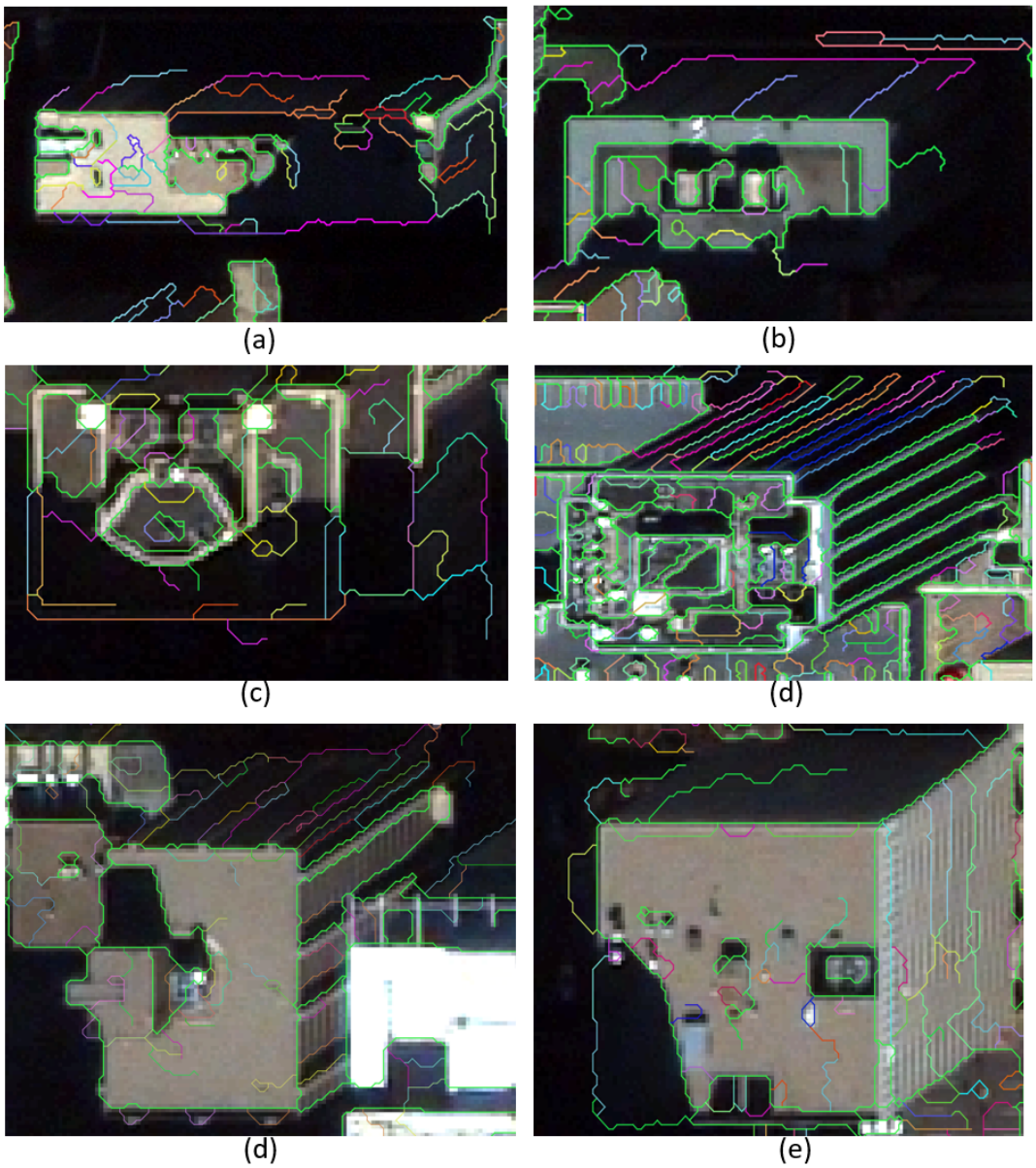


Figure 85: Edge detection results comparison under difficult conditions. Top left and bottom right images show a building partially covered with shadows. Top right and bottom left show limited contrast on building facades. Green represents the initially detected edges at the first accumulator level, and the other colours show the edges recovered from different levels of the accumulator. ABED was able to recover some hidden edges in each images. The building form and detail is partially detected. This illustrates the capabilities of ABED under different imaging conditions. However, some edges were not recovered in the dark regions, and some edges are still missing. If shadows suppress contrast variations, the edges under shadow regions are not recovered.

The results shown in Figure 86 show the effects of salt and pepper noise encountered in satellite images. Since higher levels of the accumulator implies lower smoothing, spurious edges become more evident on the roof when recovering building edges in very dark regions. This is due to the grainy surfaces with slightly noticeable colour differences on the building roofs in the top and bottom right images. However, the boundary recovery algorithm can filter irrelevant edges once valid boundaries have been recovered. Figure 86 shows the ability to recover edges under shadows, as shown by the top and bottom left image.

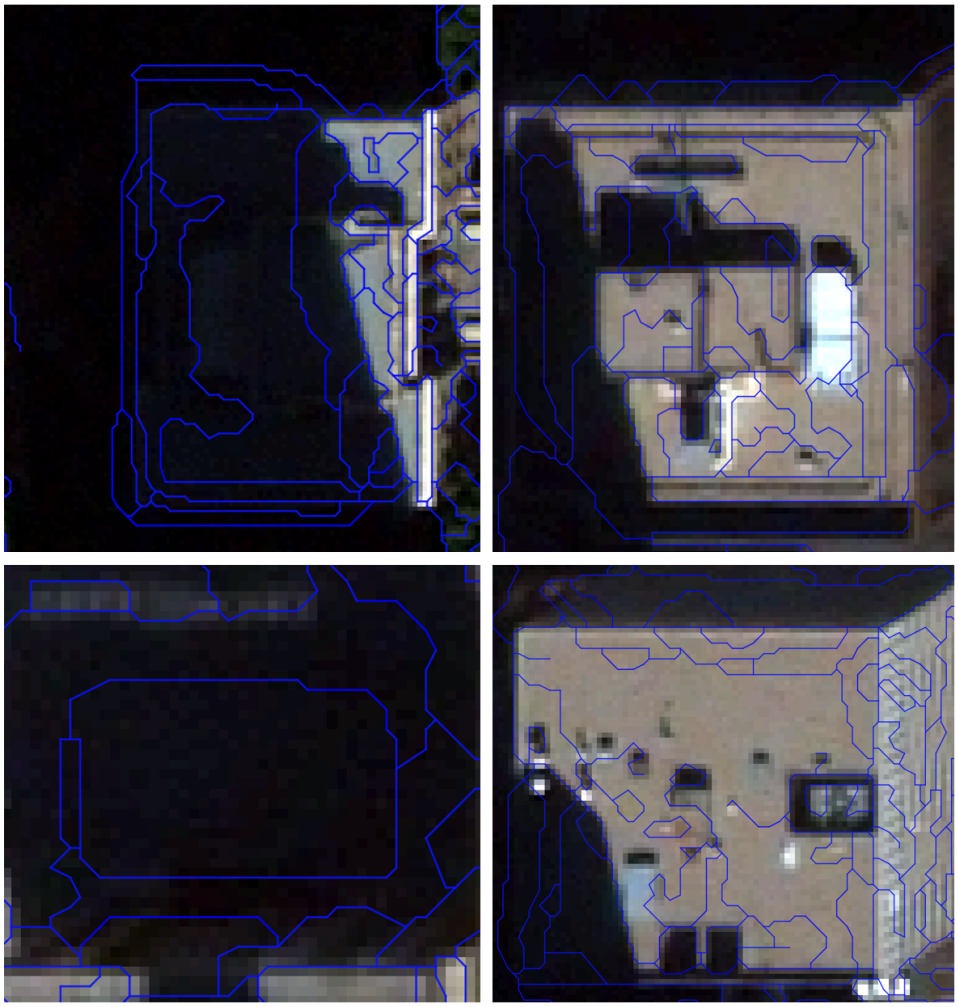
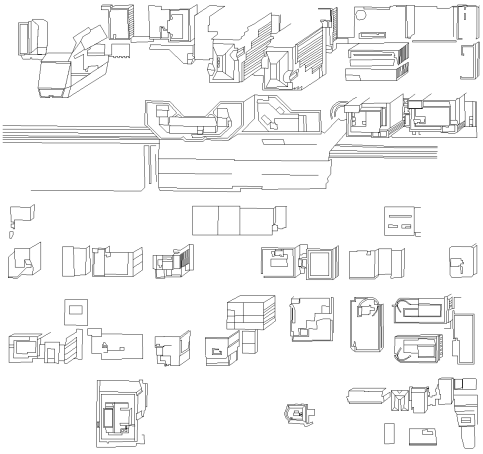
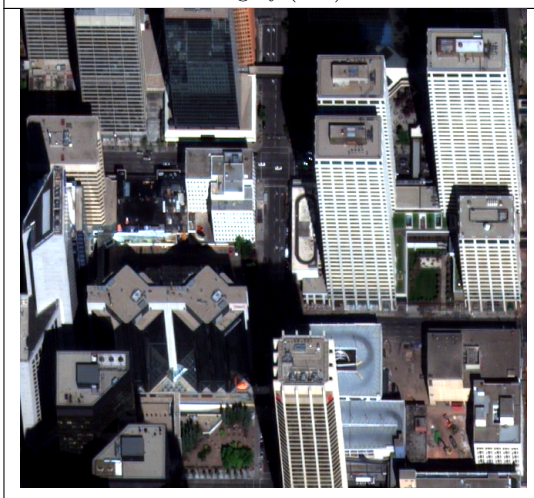
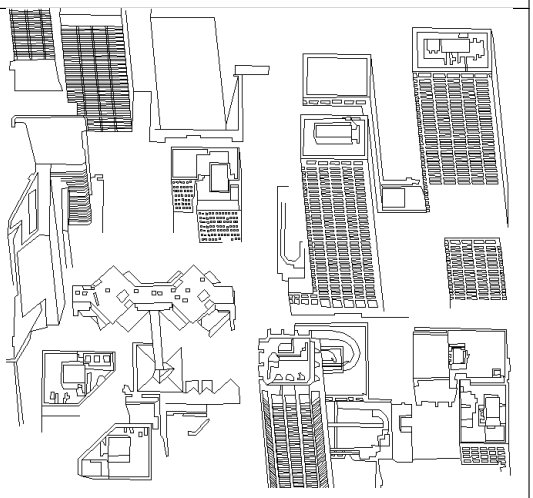
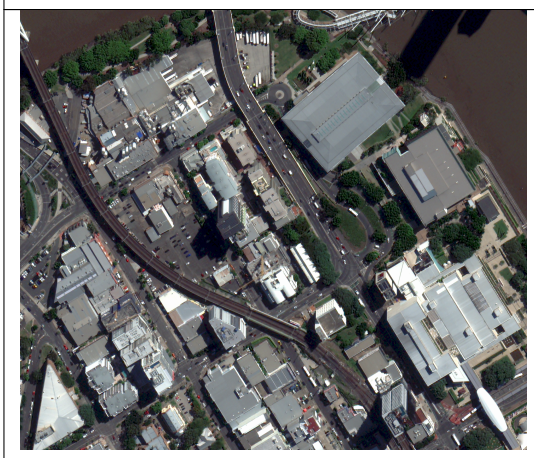
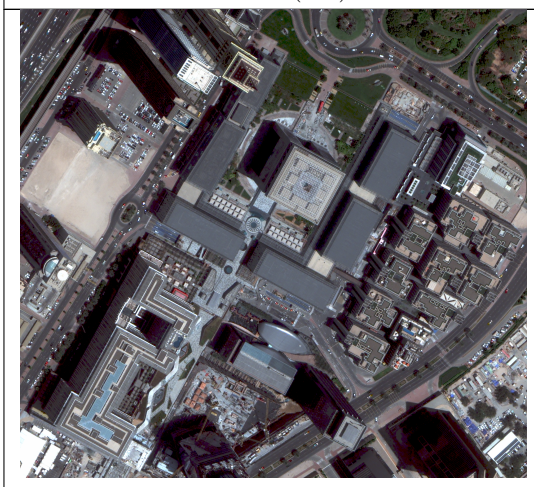
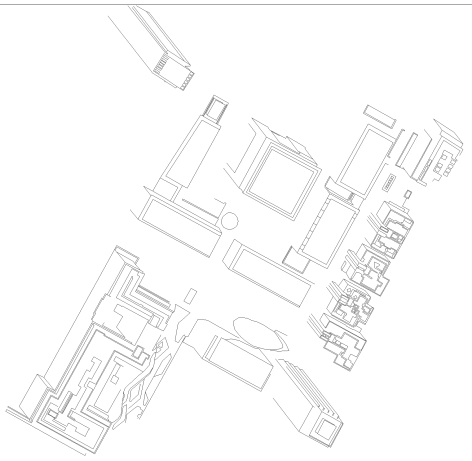
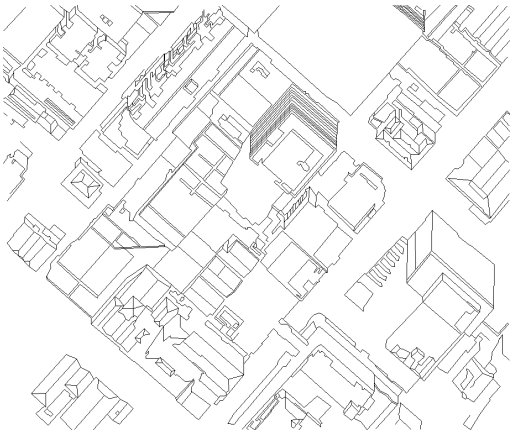


Figure 86: ABED can recover building edges that have low contrast variations. The top left and bottom left image show the building form detected in dark regions. However, Repeated patterns on the building facade, and salt and pepper noise encountered in satellite imagery images, as shown in the top right and bottom right image on the roof, can result in spurious edges. These irrelevant edges are identified during boundary reconstruction and are retained to establish valid boundary links.

7.2.2 Edge Detection and Localisation

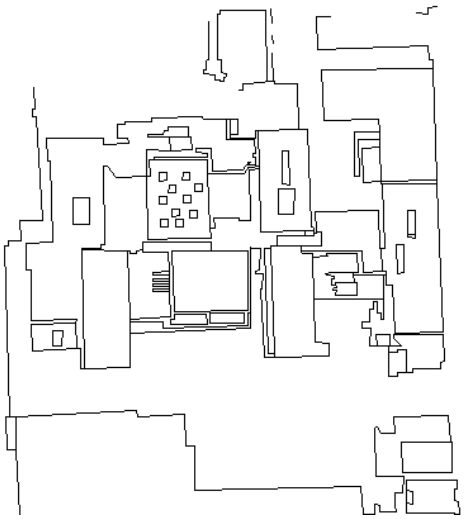
For quantitative assessment of ABED, buildings from the ten different sites were manually labelled to create ground-truth images. Table 3 shows the manually labelled ground truth images of buildings in each of the selected scenes. Where visible, the facade and roof detail were captured for comparison. Each of the edge detectors, namely EDPF, CannyPF and ABED, were compared with the ground truth images for the ten sites.

Ground Truth Images for Buildings	
	
Brisbane (TI_1)	TI_1 GT
	
Calgary (TI_2)	TI_2 GT
	
Calgary (TI_3)	TI_3 GT

Ground Truth Images for Buildings	
	
Brisbane (TI_4)	TI_4 GT
	
Dubai (TI_5)	TI_5 GT
	
Brisbane (TI_6)	TI_6 GT

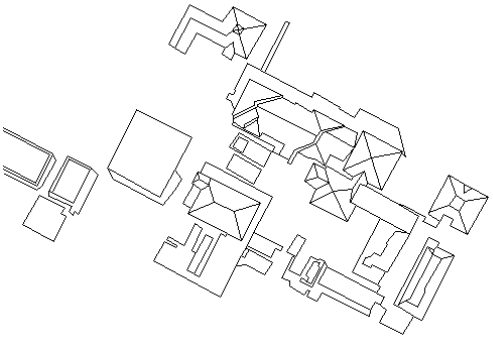
Ground Truth Images for Buildings

Calgary (TI_7)



TI_7 GT

London (TI_8)



TI_8 GT

Mourmelon (TI_9)

TI_9 GT

Ground Truth Images for Buildings

London (TI_{10})

TI_{10} GT

Table 3: Manually captured and labelled Ground Truth GT building for the 10 selected sites. The left column shows the images, and the right column show the manually labelled ground truth images.

The first objective was to establish if different gradient operators can be used in edge growth while maintaining good localisation. Figure 36 shows the effects of using different gradient operators with different parameters. The figure shows double edge effects and noisy edges. To solve for this, a localisation filter was developed to maintain good edge localisation and keep relevant edges during edge aggregation. Resultantly, two quantitative results are presented. The first results are for edge localisation. This is to verify if ABED can maintain good localisation while growing edges. The localisation comparison assesses the RMSE of the difference in edge location in all three edge detectors against ground truth images. The second set of results assesses the edge detector performance against ground truth images by computing recall. These results are tabulated in table 4.

The RMSE for all three detectors against the ground truth was computed. For all edge pixels in the ground truth, the closest pixel from each of the detectors was used to compute the euclidean distance δ_d in pixel space. After the RMSE computation, the recall was calculated using pixels that have a δ_d of one pixel or less. In addition to the computed recall and RMSE, the number of valid pixels selected as valid edges was recorded and presented in Table 4. In the table, P_n represents the total number of pixels in the GT image, D_n represents the number of valid edge pixels for each detector and M_p are edge pixels not detected. P_r is the precision of the results. Precision for ABED cannot

be computed accurately in this research. This is because ABED detects all edges in the scene. However, Ground Truth images contain edges for buildings only. Consequently, precision was computed using building Ground Truth in the next Section after building boundary recovery, and recorded in Table 7.

Edge Detection Evaluation								
Image	*Operator*	P_n	D_n	M_p	P_r	$RMSE$	*Recall*	$SD(\sigma)$
TI_1	CPF	72775	37775	35000	0.63	1.25	0.52	0.65
	EDPF	72775	37112	35663	0.68	1.28	0.51	0.64
	ABED	72775	57578	15197	0.22	1.20	0.79	0.65
TI_2	CPF	56823	28896	27927	0.45	1.27	0.51	0.68
	EDPF	56823	31241	25582	0.42	1.28	0.55	0.67
	ABED	56823	44956	11867	0.16	1.18	0.79	0.67
TI_3	CPF	48274	23996	24278	1.20	1.32	0.50	0.69
	EDPF	48274	24076	24198	1.32	1.33	0.50	0.67
	ABED	48274	34643	13631	0.51	1.25	0.72	0.68
TI_4	CPF	58011	42271	15740	0.20	1.18	0.73	0.60
	EDPF	58011	34599	23412	0.21	1.27	0.60	0.62
	ABED	58011	47558	10453	0.07	1.22	0.82	0.63
TI_5	CPF	61630	40403	21227	0.18	1.11	0.66	0.65
	EDPF	61630	34207	27423	0.18	1.20	0.56	0.65
	ABED	61630	48146	13484	0.07	1.15	0.78	0.64
TI_6	CPF	33693	23111	10582	0.49	1.12	0.69	0.60
	EDPF	33693	17310	16383	0.62	1.21	0.51	0.62
	ABED	33693	26486	7207	0.21	1.17	0.79	0.61
TI_7	CPF	10059	5268	4791	0.83	1.28	0.52	0.64
	EDPF	10059	5700	4359	0.77	1.29	0.57	0.63
	ABED	10059	7843	2216	0.25	1.20	0.78	0.65
TI_8	CPF	6019	4053	1966	0.36	1.15	0.67	0.62
	EDPF	6019	3398	2621	0.35	1.22	0.56	0.63
		6019	4619	1400	0.13	1.15	0.77	0.65
TI_9	CPF	26711	17064	9647	0.42	1.17	0.64	0.62
	EDPF	26711	13357	13354	0.46	1.26	0.50	0.62
	ABED	26711	21999	4712	0.16	1.20	0.82	0.63
TI_10	CPF	28733	13863	14870	0.23	1.28	0.48	0.61
	EDPF	28733	11652	17081	0.21	1.35	0.41	0.61
	ABED	28733	22476	6257	0.09	1.25	0.78	0.62

Table 4: Comparison of EDPF, CannyPF and ABED against GT images. The RMSE of the edge position, recall and standard deviation, σ, of the edge position were computed for each scene. ABED shows an RMSE of less than 1.4 pixels and is consistent with the RMSE of exiting detectors. ABED shows the improved recall of approximately 0.78 with a minimum of 0.72, with the other detectors reporting recall of below 0.7. ABED has the most number of pixels P_n detected, and the best recall for all sites.

A plot of the RMSE values is shown in 87. As seen from the plot, the variation in the edge position for the three detectors are all less than 1.4 pixels, which shows ABED maintains good localisation as part of the first objective. 1.4 pixels is chosen since it is the maximum displacement allowed between diagonal edges of direction 45^o on an image with regards to edge position. Edge position differs by at most one pixel because of how gradient operators identify edge pixels that have a horizontal or vertical direction. If an edge has an orientation of 45^o, this distance becomes $\sqrt{2}$, which is approximately 1.4 pixels. From table 4, all edge detectors display similar localisation errors compared to ground truth edges. Furthermore, improved recall values were reported by ABED. In all

cases, ABED had the most edge pixels compared to GT images of building edge pixels, as shown in Table 4 and plotted in Figure 88.

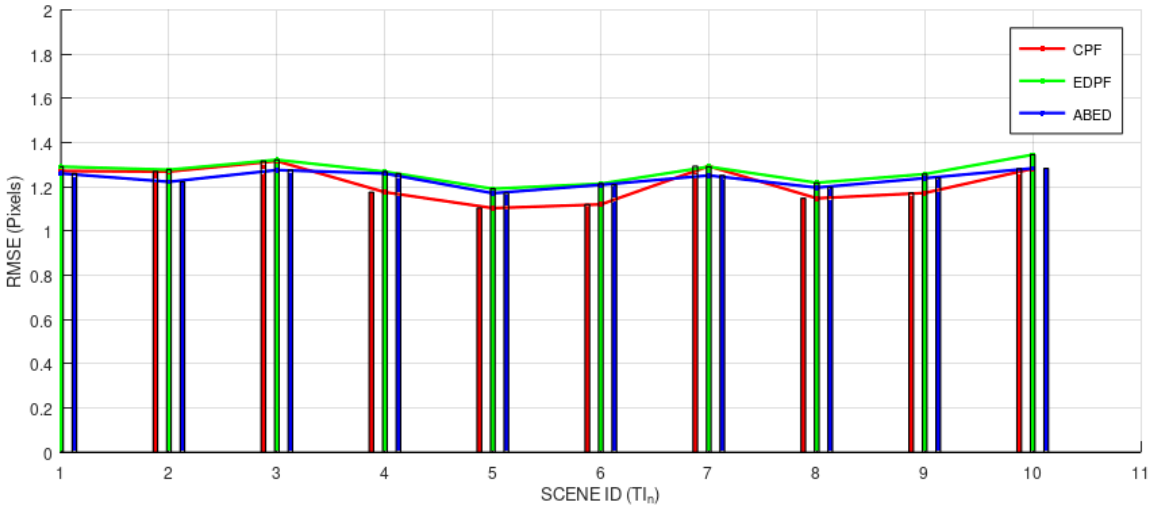


Figure 87: RMSE evaluation for edge position against ground truth edge positions. All three detectors show good localisation of edges with minimal variation across scenes. The RMSE error is below $\sqrt{2}$ or approximately pixels, which is the maximum displacement allowed for an edge pixel position along a diagonal edge.

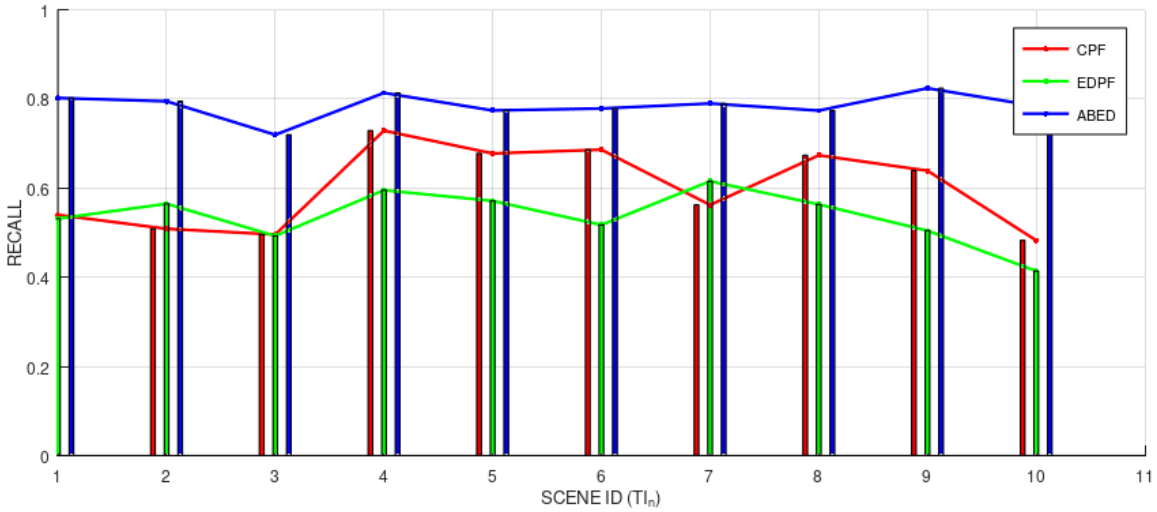


Figure 88: Recall curves for the three edge detectors namely EDPF, CannyPF and ABED. The recall for ABED improved for all ten sites by using automated parameter tuning. The recall for ABED in blue was approximately 0.8 for all sites whilst it varied with the other two detectors. This is because EDPF and CannyPF perform differently in varying image contexts. This is what ABED set out to solve.

7.2.3 Edge Growing

To further assess the amount of edge growth achieved by ABED, recall for accumulator levels was calculated from the first accumulator level to level 153. The results are shown in Table 5 where the recall values of the accumulator levels were tabulated. L_n represents the accumulator level. This was done to illustrate how the accumulator levels contribute to growing edges. The recall rate improved as more accumulator levels were processed.

Accumulator Level Recall										
$IMAGE$	L_0	L_{27}	L_{36}	L_{45}	L_{63}	L_{81}	L_{99}	L_{117}	L_{135}	L_{153}
TI_1	0.65	0.66	0.68	0.69	0.70	0.73	0.75	0.77	0.78	0.80
TI_2	0.70	0.71	0.72	0.73	0.74	0.75	0.77	0.77	0.78	0.79
TI_3	0.67	0.67	0.68	0.69	0.69	0.69	0.70	0.71	0.71	0.71
TI_4	0.70	0.71	0.72	0.74	0.74	0.76	0.78	0.79	0.80	0.81
TI_5	0.66	0.67	0.68	0.70	0.71	0.72	0.74	0.75	0.76	0.77
TI_6	0.64	0.65	0.66	0.68	0.69	0.71	0.73	0.74	0.76	0.77
TI_7	0.73	0.74	0.75	0.75	0.76	0.77	0.78	0.78	0.78	0.78
TI_8	0.67	0.67	0.69	0.70	0.71	0.73	0.74	0.75	0.76	0.77
TI_9	0.72	0.73	0.75	0.76	0.76	0.78	0.79	0.80	0.81	0.82
TI_10	0.68	0.69	0.71	0.72	0.72	0.74	0.75	0.76	0.77	0.78

Table 5: Accumulator rate of detection and edge growth for ABED. The recall for some of the accumulator levels $L_{0,27,36,45,63,81,99,117,135,153}$ for the ten sites, were selected for tabulation. The recall improved as the accumulator levels increased. This shows the ability of ABED to recover more edged using automated parameter tuning.

The results in Table 5 were plotted and shown in Figure 89. Of note are results of TI_3 and TI_7, where the edges initially recovered are good edges, the growth rate is therefore very low, indicating that the accumulator only adds relevant edges and avoids adding spurious edges using the validation function.

From Table 4, the recall value for ABED is consistently higher compared to the other detectors. Furthermore, the edges were shown to grow with aggregation using the ABED framework, which was the goal of the first objective.

7.3 ABEDL Results

The previous section presented qualitative and quantitative results of ABED. From ABED results, lines and curves require detection and reconstruction, and boundary shape reconstruction. For this, a building boundary shape reconstruction framework was developed. The framework works by detecting connected piecewise linear segments that form object form and detail, or, in other words, building boundaries. ABED was developed to provide a dense edge map to use for boundary detection and

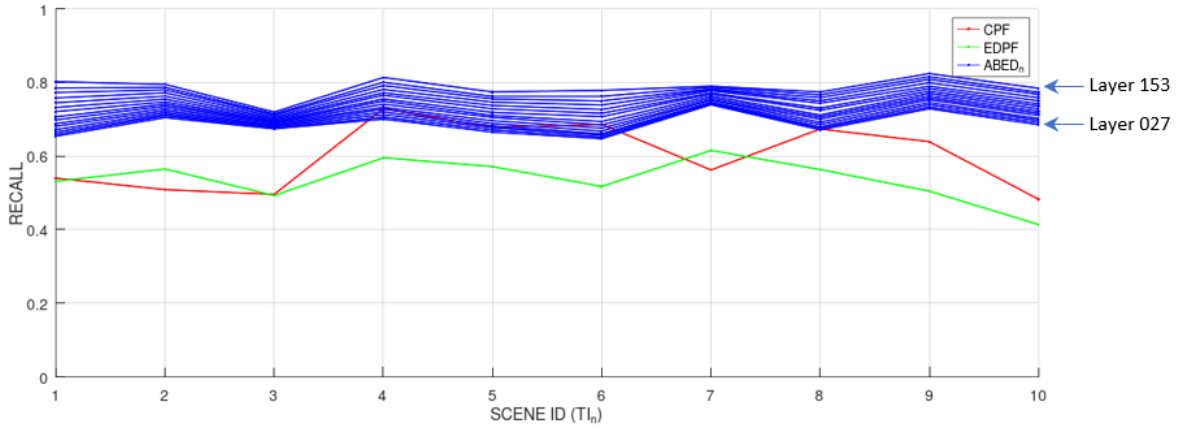


Figure 89: Plots of the recall values computed at different accumulator level tabulated in table 5. The plots show increased detection rate with increase in accumulator levels for all sites. Where initial edges are good as with TI_3 and TI_7, the growth rate is low which shows ABED add minimal irrelevant edges.

reconstruction. From the dense edge map, ABEDL recovers lines. Curves are assumed to be made up of connected line segments of varying length and smoothly varying angles at points of connection. This section presents the line detection and reconstruction results of ABEDL developed in this research, detailed in Chapter 5.

First, line classification results are shown on a larger scale. As discussed previously, there are two types of lines, that is, lines that belong to D_s) where $D_s = [0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}]$. These directions represent the horizontal, vertical, and two diagonal directions. The second type of lines have arbitrary directions whose directions are not part of D_s. From these two types, four classes were defined for boundaries namely lines that belong to D_s, lines that do not form part of D_s, curves and unclassified edges or links. Figure 90 and 91 show the dense edge map classified into these four classes using the edge graph segmentation technique developed.

7.3.1 Line and Curve Detection (ABEDL)

As mentioned in Chapter 5, unclassified lines provide links to valid linesets of different directions that form connected boundaries. This is illustrated in Figure 92.

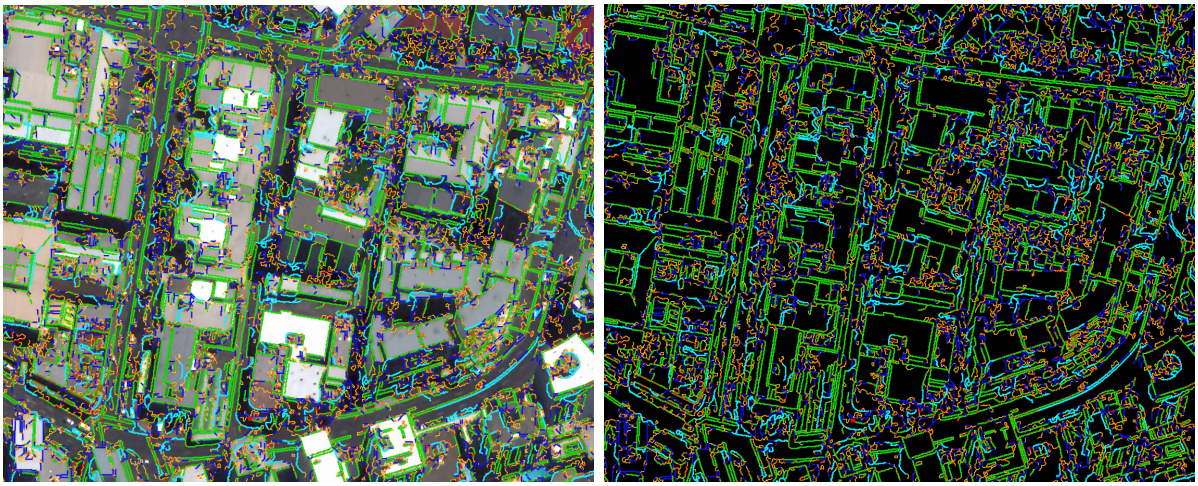


Figure 90: Classified edges from Brisbane dense edge map. Left image shows the edges overlaid with the input images. Right image shows only the edges. Navy blue shows linesets of lines that belongs to D_s, green shows linesets for lines that that do not belong to D_s, light blue linesets belong to continuous curves and the rest are unclassified edge pixels that are outliers with regards to buildings boundaries. However, the unclassified lines are used as links to connect the first three classes to form closed regions.

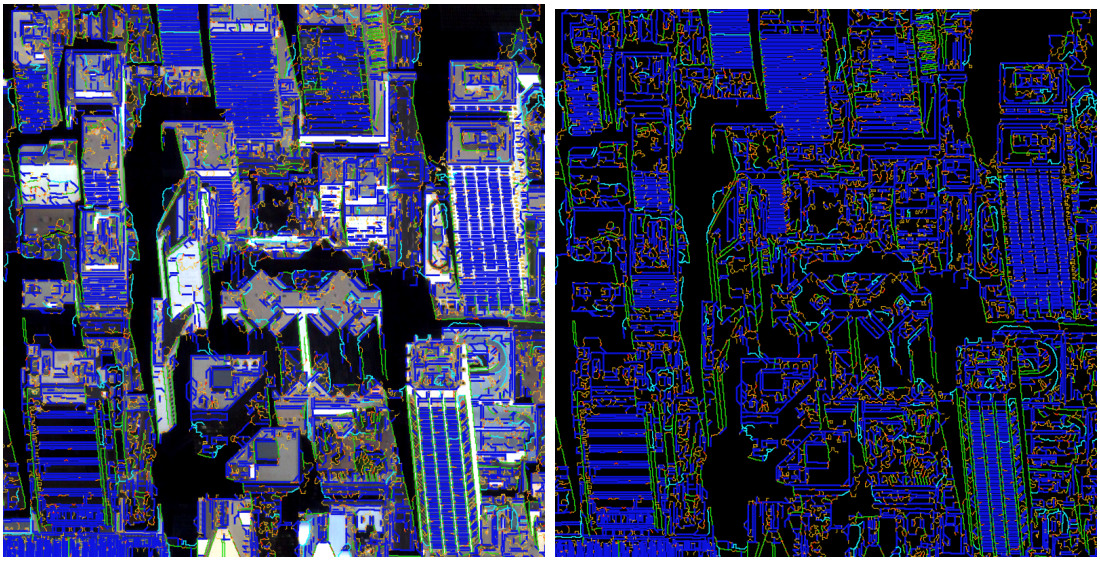


Figure 91: Classified edges from Calgary dense edge map. Left image shows the edges overlaid with the input images. Right image shows only the edges. Navy Blue shows linesets of lines that belongs to D_s, green shows linesets for lines that that do not belong to D_s, light blue linesets belong to continuous curves and the rest are unclassified edge pixels that are outliers with regards to buildings boundaries. However, the unclassified linesets are used as links to connect the first three classes to form closed regions.

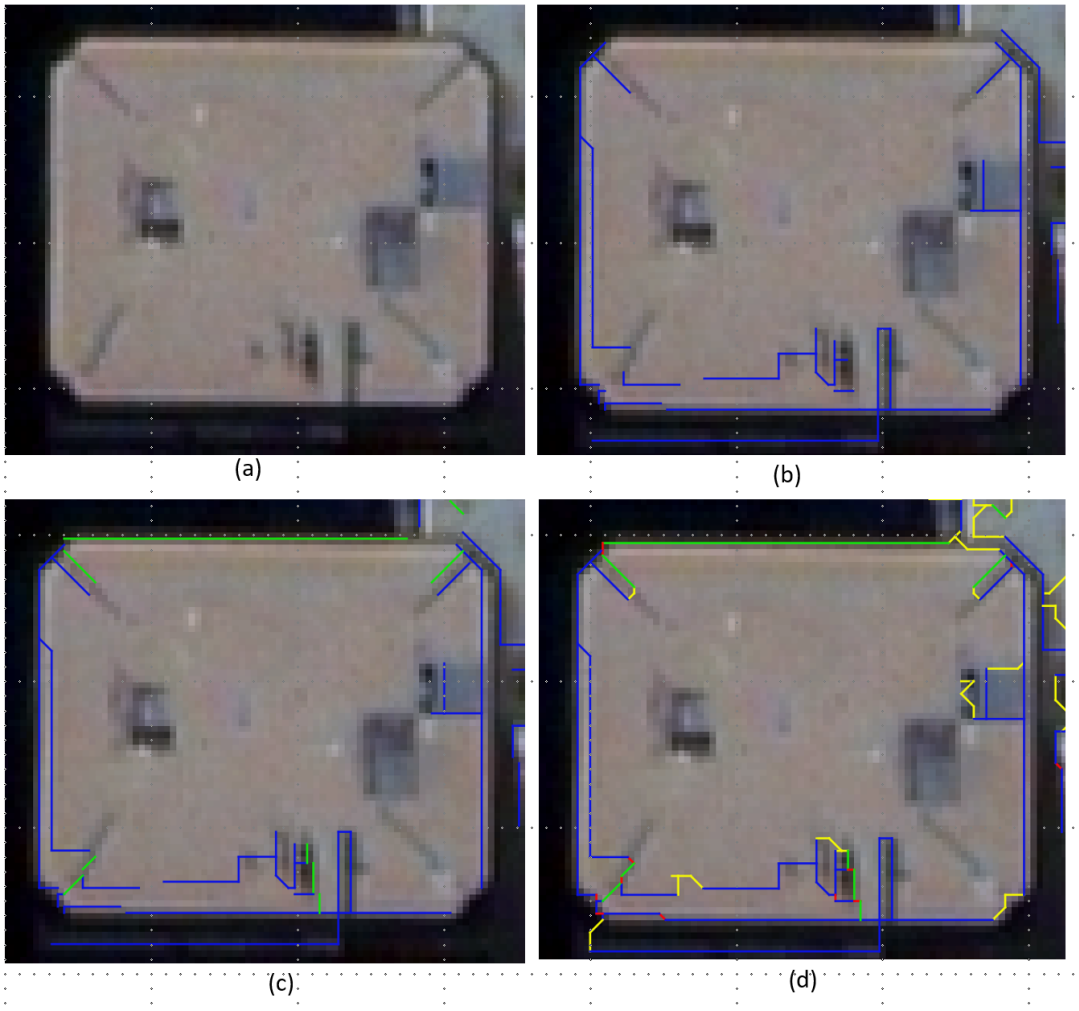


Figure 92: Classified edges from a building showing detected links in yellow and red. Navy Blue lines show linesets that belong to D_s, green lines show lines that do not belong to D_s. Image (a) shows the building, image (b) shows the lines that belong to D_s, image (c) shows lineset that do not belong to D_s and image (d) shown the links connecting the boundaries.

7.3.2 Line Reconstruction

The line detection and recovery method was presented in Chapter 5. Line detection forms the basis for boundary recovery since boundaries are visually defined as a set of connected lines. From the classified linesets, the actual lines were recovered using a breadth-first traversal method. This section presents both qualitative and quantitative results of the line detection and reconstruction method. Figure 93 shows a qualitative comparison of eight line detectors used in evaluating the developed method. Five state-of-the-art traditional line detectors, EDLines, LSD, PPHT, CannyLines, AG3Lines and three state-of-the-art deep learning methods are used for comparison, namely F-Clip, L-CNN and

HT-HWAP are used for comparison. More qualitative results are shown in Appendix A.

The first quantitative assessment compares the number of line segments detected by each algorithm versus the ground truth images created. Lines recovered by each method were matched to ground truth images using a graph based one to many mapping approach. This allowed the matching of fragmented lines that belong to one segment in the ground truth, since the ground truth images are not fragmented. Each segment was matched based on the overlap coverage of line segments whose end points lie within 1.4 pixels and deviation angle of 3^o from the matching ground truth lines. This value of 1.4 pixels is based on the average localisation RMSE error. Table 6 shows the number of lines detected for each site compared to ground-truth images. Fragmented lines that match a single ground truth line segment were counted as a single line to avoid a high count as a result of fragmentation. ABEDL detected the most lines for all ten sites, as shown by Table 6, 7 and Figures 95 and 96.

Number of Detected lines and curves										
Image	*GT*	*EDLines*	*LSD*	*PHHT*	*CL*	*AG3Lines*	*ABEDL*	F_{CLIP}	*LCNN*	*HTHAWP*
TI_1	3798	1583	1409	25	1063	1236	2398	46	206	254
TI_2	2370	1105	986	239	895	1022	1773	241	283	216
TI_3	4091	1637	896	218	952	1688	2653	259	558	534
TI_4	2459	1252	1136	94	962	1094	1774	9	82	124
TI_5	1989	1066	949	66	820	963	1457	5	43	98
TI_6	2081	878	830	199	692	765	1330	32	255	314
TI_7	602	235	234	52	195	235	383	110	243	242
TI_8	348	172	158	39	157	144	238	23	50	180
TI_9	1585	699	633	51	485	567	1083	20	108	211
TI_10	1400	422	361	69	346	342	942	8	43	104

Table 6: Number of detected lines for the eight detectors versus ground truth for each of the ten samples. ABEDL detected the most lines for all ten sites compared to other detectors. While other detectors returned considerable number of lines, PPHT, F-Clip, L-CNN and HT-HWAP detected very few lines. This is because PPHT does not perform well without parameter tuning, while F-Clip, L-CNN and HT-HWAP, which are based on deep learning, were trained on terrestrial images that do not capture the satellite scene contexts. This highlights the challenges with deep learning methods with regards to adapting to different contexts without the need for re-training. However, L-CNN performed better on TI_7 because the scene was made of a single building, which resembles that deep learning training contexts, of large single buildings using terrestrial images.

Assessing the quality and quantity of individual lines has not been fully addressed in literature [1, 6, 3]. Most research assesses the quality and quantity of lines as a group. Most studies evaluate line detection using either the number of lines detected, recall, or precision. The challenge of using recall and precision is that it depends on what exists in the ground truth and what is deemed true positive by the application. For example, if a large number of valid lines are returned in comparison to ground truth images that

Figure 93: Line detection results comparison for Brisbane CBD (TI_1). ABEDL has denser lines against five detectors namely EDLines, LSD, PPHT, CannyLines and AG3Lines. The lines represent both form and detail, which is absent in some of the detectors. GT is the ground truth image.

have a different set of lines, then precision is low, even though all lines returned are valid. Table 7 shows the precision and recall values for the comparison. Recall may be relied on, however, precision is not a reliable measure since ground truth did not contain all valid lines recovered. A better recall is recorded for the lines detected by ABEDL. PPHT detected very few lines as reported by a precision of one, but poor recall, since most lines are not detected. This is because PPHT requires parameter tuning. However, the optimal parameters can only be established by experimentation. Three state-of-the art deep learning method benchmarked namely F-Clip, L-CNN and HT-HWAP, failed to recover most of the lines from satellite images because the models are not trained with satellite image data. These results are shown in Figure 94. This highlights the challenges with deep learning methods with regards to adapting to different contexts without the need for re-training. The test was performed on existing deep learning models using open-source code. The models were used as provided by the authors. For this reason, the evaluation was only limited to a quantitative evaluation of existing deep learning methods. For deep learning methods to perform, the training data must represent the data in the input images. Furthermore, Deep learning methods do not adapt to image resolution changes [141]. However, for scene TI_7, deep learning methods were able to match the performance of EDLines, LSD, PPHT, CannyLines and AG3Lines with regards to the number of lines detected. This is because scene TI_7 consists of a single building, which reflects the geometry represented in the training data [86, 87].

To quantify the detection differently, a one-to-one comparison of line segments was carried out where the orientation and line coverage was computed. Using line orientation, matched lines were projected onto ground truth lines and coverage was computed as the detected length (DL) as a percentage. Effectively, full coverage of a line is achieved if both orientation and length are accurate, while less coverage is recorded for less accurate and fragmented lines. The matched segments were sorted according to coverage for each site and the graphs are shown in Figures 95 and 96. Put differently, the recall and precision for each line were represented as coverage. The longer blue curves indicate that for all sites, more lines, which are longer in comparison to the benchmarked methods, were detected. However, the decay of the curves towards the end indicates that some lines are not fully detected. This happens because fragmentation is not eliminated but reduced, and some edges are still missing. Since the amount of fragmentation and missing scenes vary across

Image	P	R	$F1_{score}$	Fr	P	R	$F1_{score}$	Fr	P	R	$F1_{score}$	Fr
	EDLines				LSD				PPHT			
TI_1	0.93	0.54	0.69	0.058	0.94	0.48	0.63	0.08	1.0	0.01	0.01	0.00
TI_2	0.89	0.61	0.72	0.093	0.88	0.55	0.68	0.1	0.96	0.09	0.16	0.02
TI_3	1.0	0.62	0.76	0.027	1.0	0.42	0.59	0.028	1.0	0.06	0.12	0.00
TI_4	0.86	0.67	0.75	0.094	0.89	0.61	0.72	0.11	0.98	0.04	0.07	0.01
TI_5	0.89	0.7	0.79	0.1	0.91	0.59	0.72	0.14	1.0	0.02	0.04	0.0
TI_6	0.93	0.6	0.73	0.072	0.93	0.56	0.7	0.08	0.99	0.11	0.2	0.02
TI_7	0.92	0.6	0.73	0.076	0.95	0.56	0.71	0.08	0.99	0.08	0.15	0.02
TI_8	0.95	0.62	0.75	0.049	0.98	0.58	0.73	0.07	1.0	0.09	0.16	0.01
TI_9	0.88	0.54	0.67	0.082	0.9	0.47	0.62	0.09	0.99	0.026	0.01	0.01
TI_10	0.92	0.37	0.52	0.051	0.93	0.28	0.43	0.07	0.95	0.03	0.06	0.01
	CannyLines				AG3Lines				ABEDL			
TI_1	0.96	0.43	0.59	0.031	0.92	0.49	0.64	0.034	0.86	0.66	0.75	0.15
TI_2	0.9	0.53	0.67	0.068	0.85	0.62	0.72	0.076	0.8	0.75	0.77	0.27
TI_3	1	0.45	0.62	0.016	0.95	0.64	0.76	0.013	0.94	0.72	0.82	0.07
TI_4	0.88	0.57	0.69	0.07	0.83	0.67	0.74	0.066	0.82	0.77	0.8	0.24
TI_5	0.91	0.61	0.73	0.087	0.84	0.72	0.78	0.072	0.82	0.8	0.81	0.26
TI_6	0.93	0.54	0.68	0.037	0.92	0.61	0.73	0.032	0.85	0.72	0.78	0.21
TI_7	0.98	0.51	0.67	0.032	0.9	0.63	0.74	0.045	0.85	0.72	0.78	0.15
TI_8	0.99	0.57	0.73	0.043	0.96	0.59	0.73	0.034	0.88	0.72	0.79	0.18
TI_9	0.92	0.42	0.58	0.043	0.86	0.52	0.65	0.042	0.81	0.69	0.74	0.25
TI_10	0.9	0.31	0.46	0.039	0.87	0.37	0.52	0.024	0.82	0.53	0.65	0.24
	F-Clip				L-CNN				HT-HWAP			
TI_1	0.27	0.02	0.03	0.0	0.62	0.1	0.18	0.0	0.54	0.099	0.17	0.00
TI_2	0.32	0.15	0.20	0.01	0.73	0.19	0.3	0.00	0.62	0.13	0.22	0.0
TI_3	0.87	0.1	0.17	0.0	1.0	0.24	0.38	0.00	0.93	0.22	0.36	0.00
TI_4	0.51	0.01	0.02	0.0	0.45	0.08	0.13	0.0	0.27	0.057	0.09	0.0
TI_5	0.56	0.0	0.01	0.0	0.52	0.05	0.09	0.00	0.33	0.08	0.13	0.0
TI_6	0.74	0.03	0.05	0.0	0.72	0.2	0.32	0.00	0.53	0.24	0.33	0.01
TI_7	0.94	0.33	0.49	0.01	0.98	0.61	0.75	0.02	0.77	0.51	0.62	0.04
TI_8	0.73	0.11	0.02	0.0	0.95	0.24	0.38	0.01	0.57	0.62	0.6	0.08
TI_9	0.61	0.03	0.06	0.0	0.62	0.12	0.2	0.00	0.38	0.19	0.26	0.01
TI_10	0.31	0.01	0.02	0.0	0.62	0.061	0.11	0.0	0.25	0.088	0.13	0.01

Table 7: The performance of ABEDL with regards precision (P), recall(R), F1-Score and Fragmentation ration F_r. ABED records the best recall. However, while precision is good for ABEDL, it was outperformed by EDLines and CannyLines. This is because EDLines and CannyLines only retain salient lines but do not detect most feint lines, while abed had detail that was not present in the ground truth. However, ABEDL has the better F1-Score, highlighting the superior performance. The fragmentation ratio in ABEDL is higher because links are used to connect lines. These links are not taken into account when computing line fragmentation.

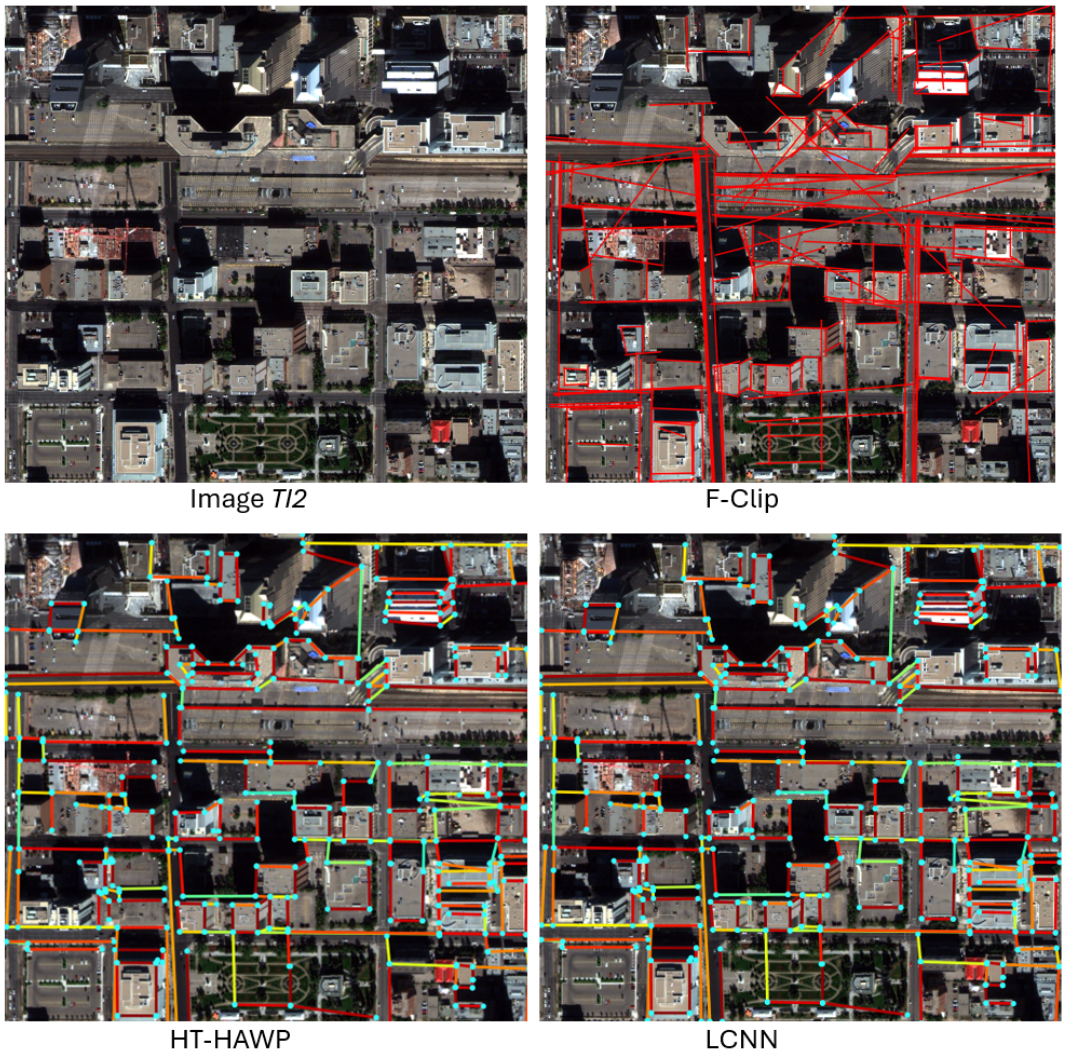


Figure 94: Deep learning based line detection results for image TI_2 of Calgary. Deep learning based methods overestimate line length and does not recover object detail. Line recovered are for large object as evidenced by F-Clip, L-CNN and HT-HWAP. However, some lines are not limited to a single building boundary but combines multiple building edges into a single line segment. The result shown above can be greatly improved by training the deep learning models with satellite image data.

scenes, the percentage length detected will consequently vary. This is reflected in the different decay shapes of the curves. However, the decay is more evident is site T_{10} which is mostly a residential area. This happens because more edges are missed because trees cover houses and small buildings with sides less than two metres. Pixel mixing on short edges blurs the small edges. However, an interesting observation to note is that the deep learning based methods detect fewer lines, but with reduced fragmentation, as evidenced by the sharp decay for all Deeep learning based methods. TI_8 is a single building object. This is the reason HT-HWAP and L-CNN, performed well in detecting lines since they

were trained for single-building scenarios.

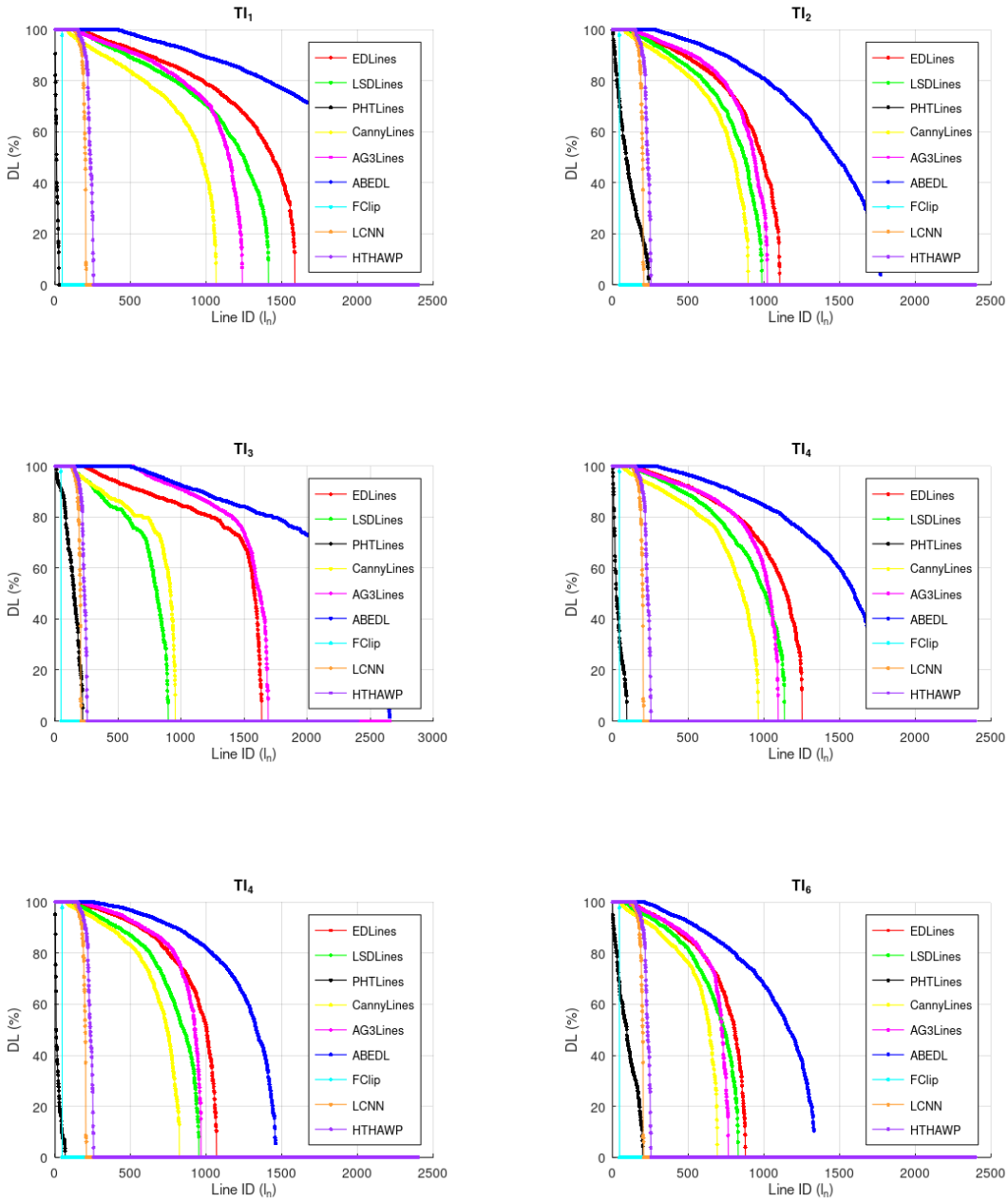


Figure 95: Line coverage comparison for each detected line versus ground truth lines. The x-axis represents the index of the line, and the y-axis shows the coverage or detected length (DL) as a percentage. ABEDL in blue shows the highest number of longer lines that were detected across these six sites. However, the decay of the blue curves at the end indicate fragmentation and missing edges since these are reduced but not eliminated. However, every line in the ground truth had a full or partial detection.

To further evaluate the level of detection and the line growth rate achieved by ABEDL, the detected lines were compared to the ground truth lines of a tennis court in Figure 97. The tennis court was selected to establish the rate of detection where lines are not

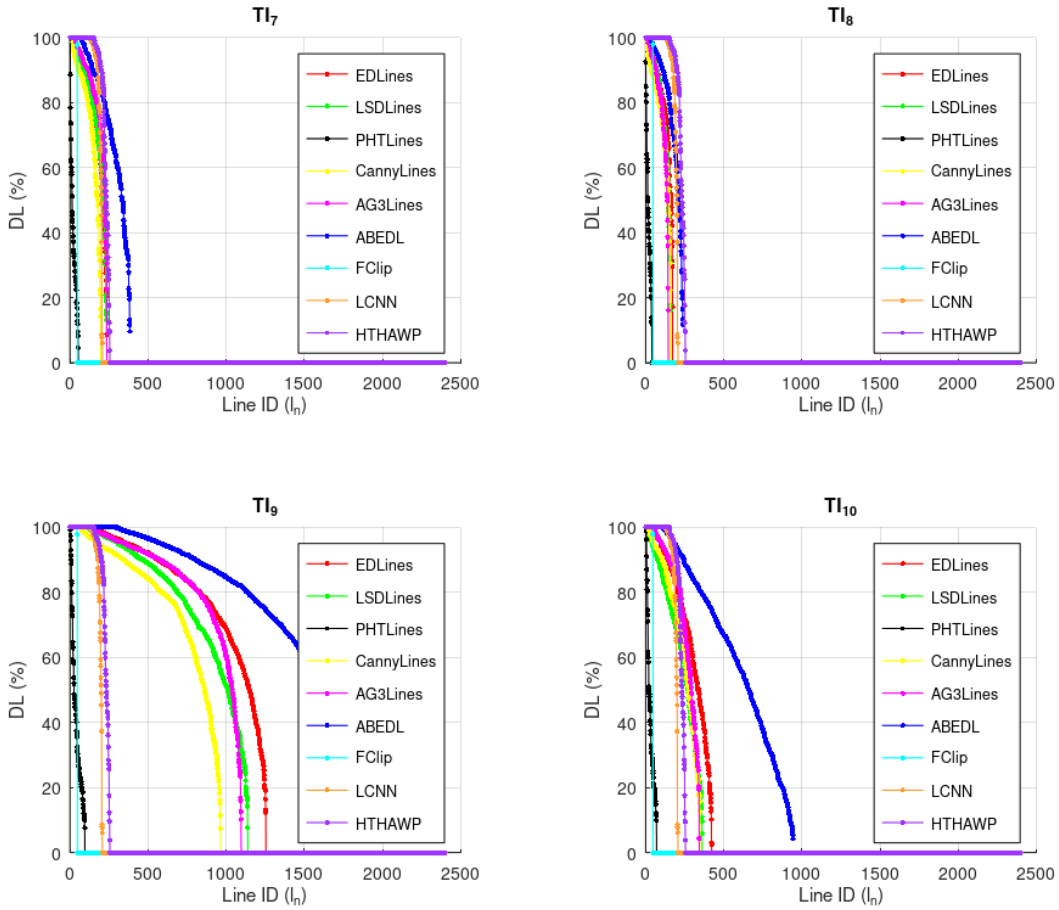


Figure 96: Line coverage comparison for each detected line versus ground truth lines. The x-axis represents the index of the line, and the y-axis shows the coverage or detected length (DL) as a percentage. ABEDL in blue shows the highest number of longer lines that were detected across the ten sites. The decay of the blue curves at the end of each indicate fragmentation and missing edges still exist since these are reduced but not eliminated. However, every line in the ground truth had a full or partial detection. Sample TI_{10} has a sharper decay since it is a residential area with smaller objects and trees. Trees and small edges contribute to missed edges and fragmentation. HT-HWAP and L-CNN, performed well in detecting lines on TI_7, which is an image of a single building. Single building images resemble the training context of current deep learning models, which use terrestrial images of single building structures.

clearly visible. The comparison was based on the evaluation of the true length of the lines versus the detected lines. The detected and ground-truth lines were paired using a proximity threshold of 1.4 pixels for line end points. Pairs were established by selecting the closest parallel edges. A one-to-many mapping was enforced to avoid fragmented lines from being treated as separate lines. From these matches, the number of detected lines per accumulator level and overall pixel coverage were computed at different accumulator levels. The results are presented in Table 8.

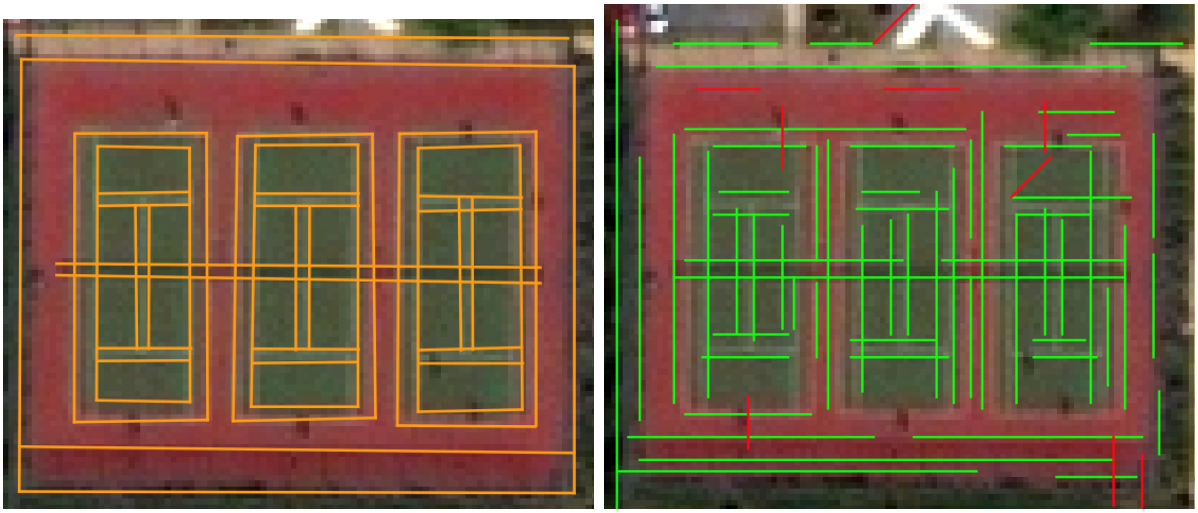


Figure 97: A comparison of Lines detected from edges from ABED vs ground truth. This figure shows the performance of ABEDL in areas of low contrast. Table 8 shows an 89% detection rate, which confirms the recall rate of approx 0.8 from Table 7. Despite the low contrast of the lines on the tennis court, every line had a partial or full detection. This was the goal of the second objective. The faint lines with the low contrast inside the tennis court were detected. Most of the lines are partially or fully detected as shown by the 89% detection rate in table 8. However, fragmentation is still evident and a few edges missed. This figure shows the result with 37% more edges detected versus EDPF as reported in table 8.

The line lengths from the ground truth image were used as the baseline. At different accumulator levels Acc_L, the number of detected lines that match ground truth lines D_L was recorded. The ratio of matched line pixel, or recall, R_L, against the baseline was calculated for each accumulator level. Next, the total recall R_T, which is the sum of all recalls for all accumulator levels, and the amount of extra pixels or the percentage growth in line length G_T from the edge-growing process at a given accumulator level. Only the accumulator levels that returned valid edges were recorded out of 232 levels. From the table, the accumulator contributed approximately 37% more line pixels from the initially detected 52%. The red edges in Figure 97 are outliers that did not match any ground

Edge Detection Evaluation				
Acc_L	DL	$R_L(\%)$	$R_T(\%)$	$G_T(\%)$
0	28	52.2219	52.2219	0
1	3	4.63325	56.8552	4.63325
2	1	0.92665	57.7818	5.5599
4	1	2.34388	60.1257	7.90378
5	1	0.872141	60.9978	8.77592
6	1	0.599597	61.5974	9.37551
7	1	0.817632	62.4151	10.1931
9	1	0.654106	63.0692	10.8473
12	1	0.872141	63.9413	11.7194
13	1	1.09018	65.0315	12.8096
18	1	1.19919	66.2307	14.0088
19	1	0.981158	67.2118	14.9899
20	1	0.709952	67.9218	15.6999
27	3	4.03468	71.9565	19.7346
28	2	1.68388	73.6403	21.4184
30	1	0.708614	74.349	22.1271
32	1	4.52454	78.8735	26.6516
33	1	2.23486	81.1084	28.8864
36	2	1.79845	82.9068	30.6849
47	2	2.77995	85.6868	33.4648
50	2	1.47174	87.1585	34.9366
206	1	0.922637	88.0811	35.8592
224	1	0.708614	88.7897	36.5678
232	1	1.19919	89.9889	37.767

Table 8: Rate of detection and line growth rate per accumulator level for the tennis court in Figure 97. This table shows the percentage of line pixels contributed by different accumulator levels. The recall R_L, for all detected line pixels, for each accumulator, is calculated. The total recall (R_T), is the sum of recall values for all accumulator levels, which increases with an increase in accumulator levels. Overall, boundary coverage increased by 37% from an initially detected boundary using EDPF. G_T shows the additional percentage of boundary pixels recovered from different levels of the accumulator. Accumulator levels (Acc_L) that returned valid lines were recorded. Approximately 89% of valid line pixels from the tennis court boundary were detected, which is desirable for boundary completion.

truth edges. An overall detection rate of 89% was achieved with the tennis court.

7.4 Building Boundary Shape Reconstruction results

The previous section presented results on the recovered lines from ABEDL using the method detailed in Chapter 6. This section presents the boundaries recovered by ABEDL. As explained in Section 6, the boundaries are recovered by two methods. The first method reconstructed boundaries by connecting linear and curved segments by unclassified edge links. Because of the dense edge network, most boundaries are recovered, and closed regions are automatically created when unclassified edge links are added. An example is shown in Figure 98. Even with boundary shape distortion due to projective distortion, the shape is recovered with ABEDL in image (a). Image (b) only recovered lines with AG3Lines detector. More boundary lines are recovered by the BFS boundary shape recovery algorithm presented in Section 6.1.5. Figure 99 below shows some recovered boundaries by edge linking. The yellow edges represent the unclassified edges that complete the boundaries by addition. The second method was to estimate the completeness of the boundaries based on Gestalt principles. In the case of missing links, proximity, continuity, similarity, and closure are used to complete boundaries. The quantitative assessment presented for lines in the previous section shows the improved boundary coverage. The boundary detection results are shown in Figures 99, 100. More figures are shown in Appendix B.

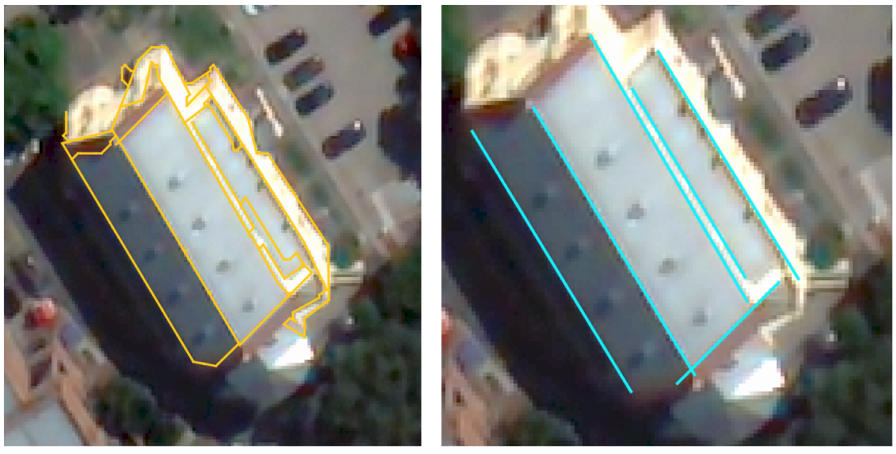


Figure 98: Recovered boundaries using ABEDL versus AG3Lines. Image (a) shows a more complete boundary shape when compared with image (b). ABEDL completes boundaries using links hence the building form is detected more compared to AG3Lines.

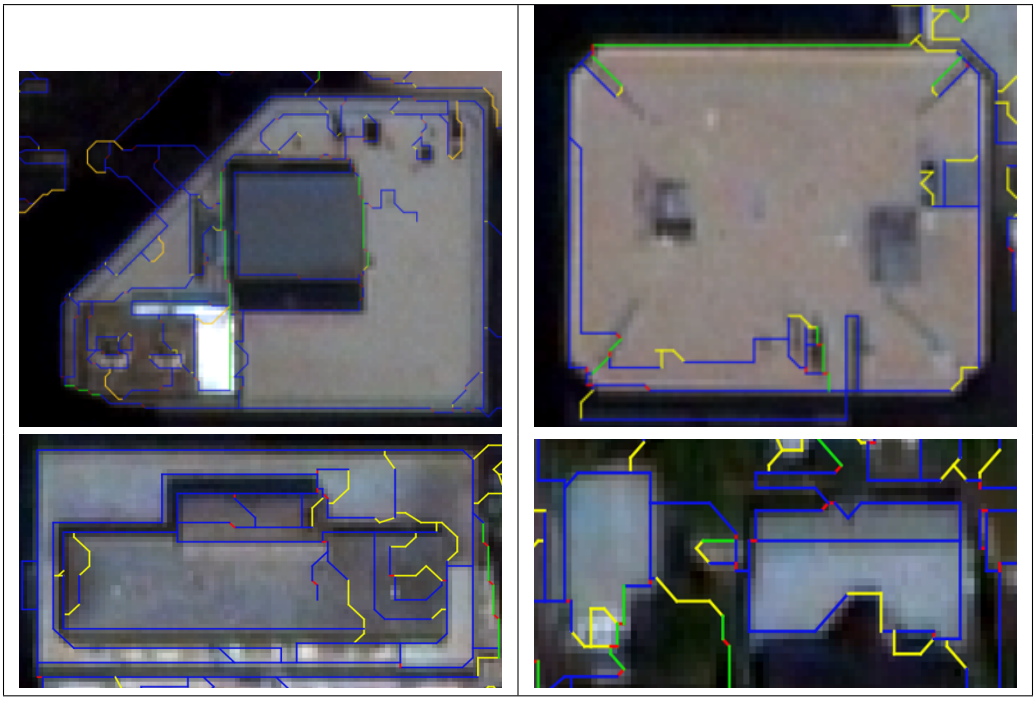


Figure 99: Boundary detection results. Boundary closures using links in yellow. All links that connect valid boundaries are retained forming closed regions in all four images.

ABED attempts to recover the detail of a small objects using links. A closer look at the boundaries shown in Figures 99 and 100, there are non-regular lines along straight boundaries. These are the recovered links for very small objects or distorted corners. A closer look at the advantages of using links is shown in Figure 101. In addition to the lines, links have completed corners that AG3Lines failed to recover. This is the advantage of using ABED and ABEDL. The benefits of having this detail are visible with facade detail in the wireframes produced in the next section. Another example of detail is shown in Figure 102. The detail of the roof recovered by ABEDL is visible. Because of the short segments, the boundary recovered exist but not visually accurate. The benefit of such detail is in the resulting 3D wireframes.

To demonstrate the advantages of maintaining a connected network, Figure 103 shows the recovered lines for a curved structure. ABED maintains line connectivity and treats line fitting as a global rather than a local problem. The red circle shows the failure to resolve line end points by AG3Lines since the lines are fit independently.

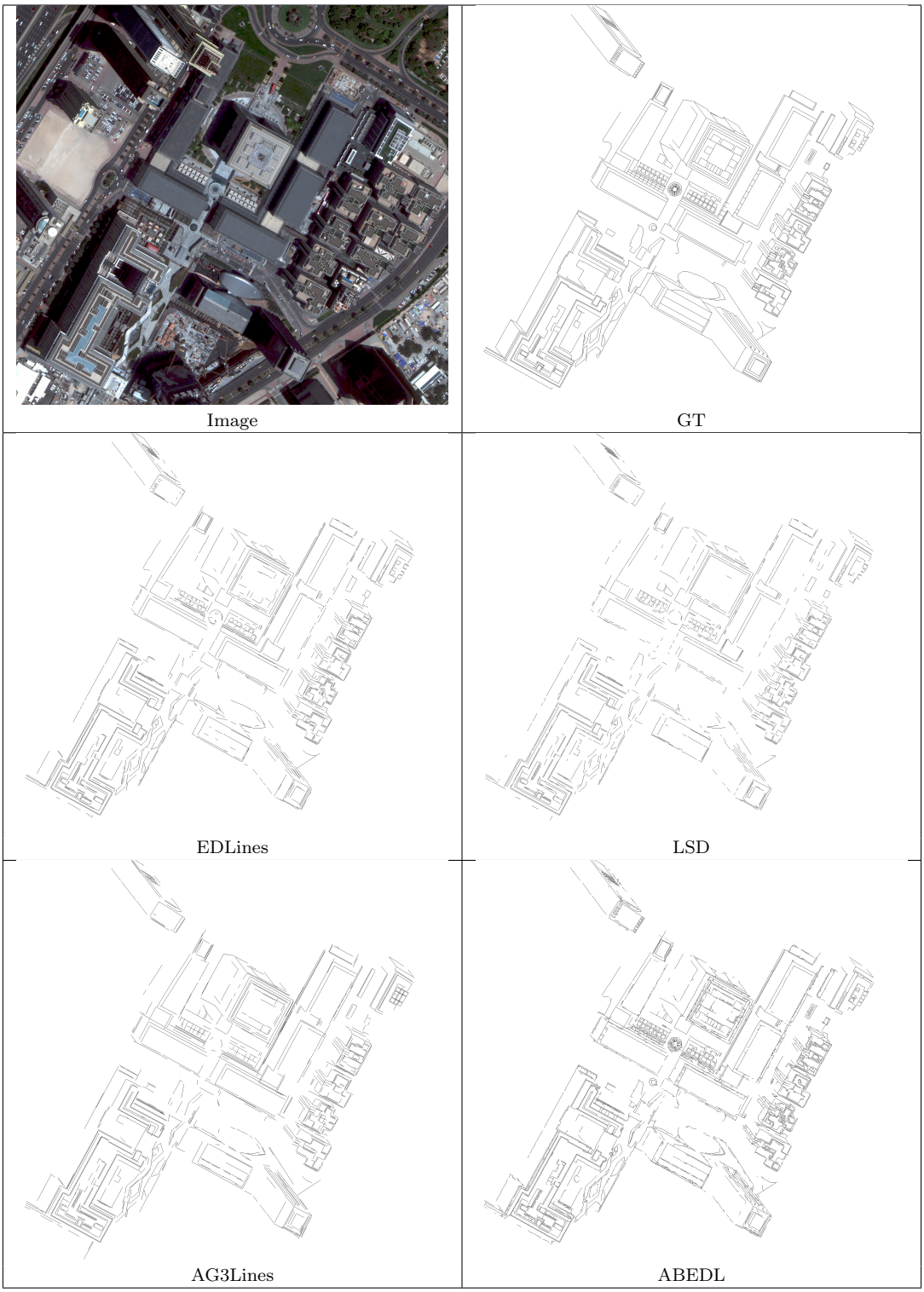


Figure 100: Boundary detection results comparison for Dubai CBD (TI_5). ABEDL, has denser and longer boundaries with roof and facade detail retained. However,the links distort the boundary shapes.

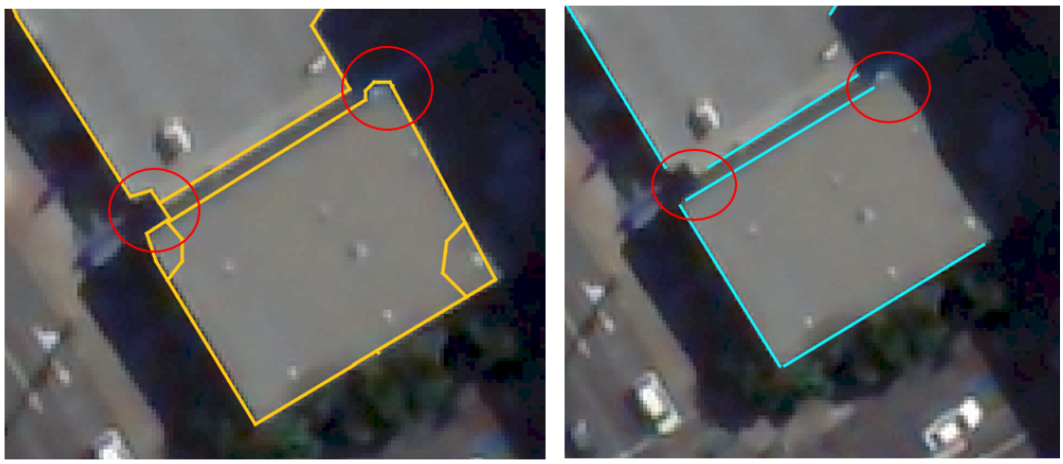


Figure 101: Recovered boundaries using ABEDL versus AG3Lines. Image (a) shows a complete building boundary shape. ABEDL completes boundaries while preserving detail.

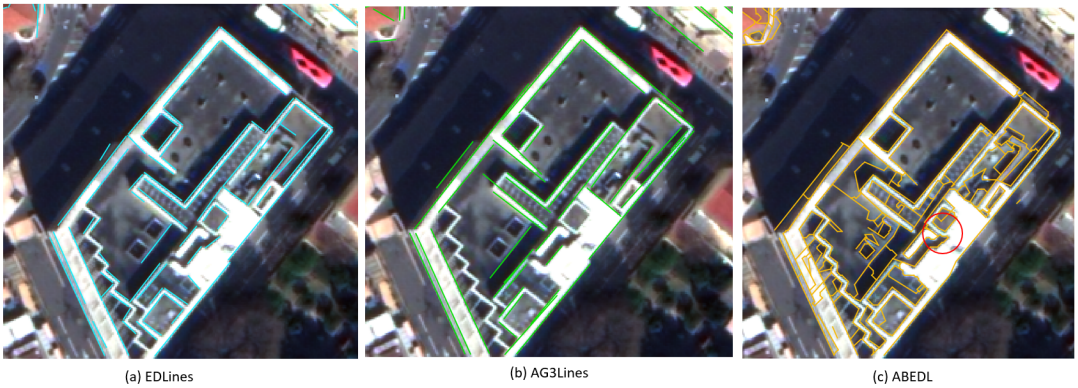
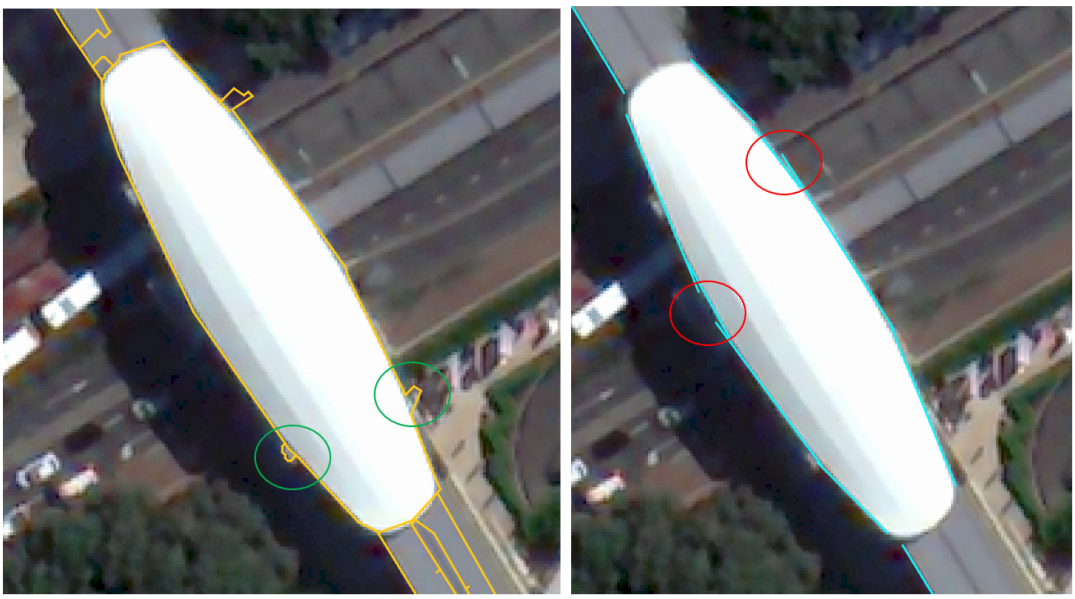


Figure 102: Recovered boundaries using ABEDL versus AG3Lines and EDLines. Image (a) shows lines recovered by EDLines, image (b) are lines from AG3Lines and image (c) are lines from ABEDL. More detail is captured on image (c), However, since the roof detail is comprised of small edges, distortion in the boundary shape of building detail is evident.



(a) ABEDL (b) AG3Lines

Figure 103: Recovered boundaries using ABEDL versus AG3Lines. Image (a) shows connected and continuous curves while image (b) has issues resolving line end points (red circles). The is as a result of local line fitting which is done independently. Kinks are visible (green circles) where ABEDL attempts to retain correct boundary shapes for small objects.

7.5 Multiview Reconstruction

The boundaries reconstructed from the over arching framework were matched, and 3D wireframes were created. Three of the ten samples were used in creating the wireframe models. The models are shown in Figures 104, 105, 106, 107 and 108.

7.6 Performance Analysis

The edge detection and building boundary shape reconstruction framework were tested on a Windows 10 machine, with 64 Gig RAM, using a single core for processing. To benchmark the performance of the framework, the time taken for edge detection was recorded separately from the boundary reconstruction time. The framework was tested with non-optimised code, that was producing outputs for each step, which increased run-time. The first to be recorded was the edge detection processing time for each of the ten sites. The time was recorded for each level in an accumulator of 153 levels. The results are plotted in Figure 109.

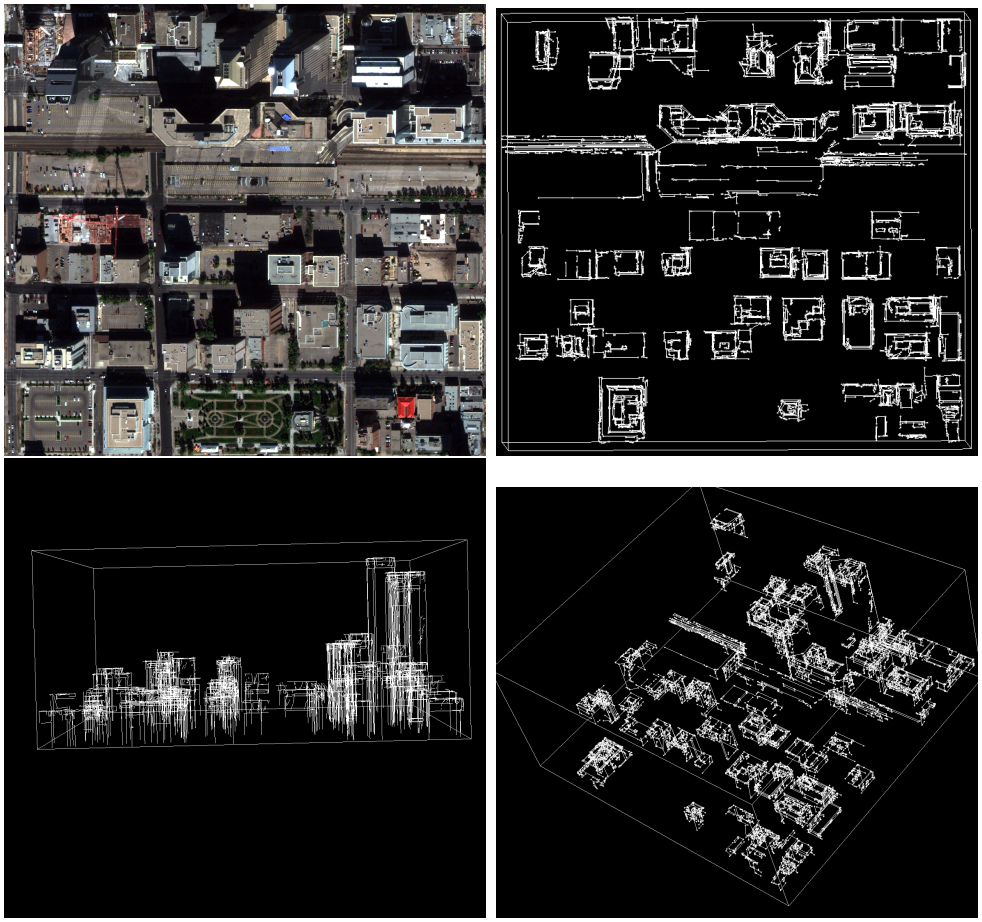


Figure 104: 3D Wireframe model of Calgary created from reconstructed boundaries by line segment matching. The top left image is on of the input image, top right shows the top view of the wireframe. The superstructures and parapet boundaries are visible. Bottom left shows the side view with some facade details reconstructed.

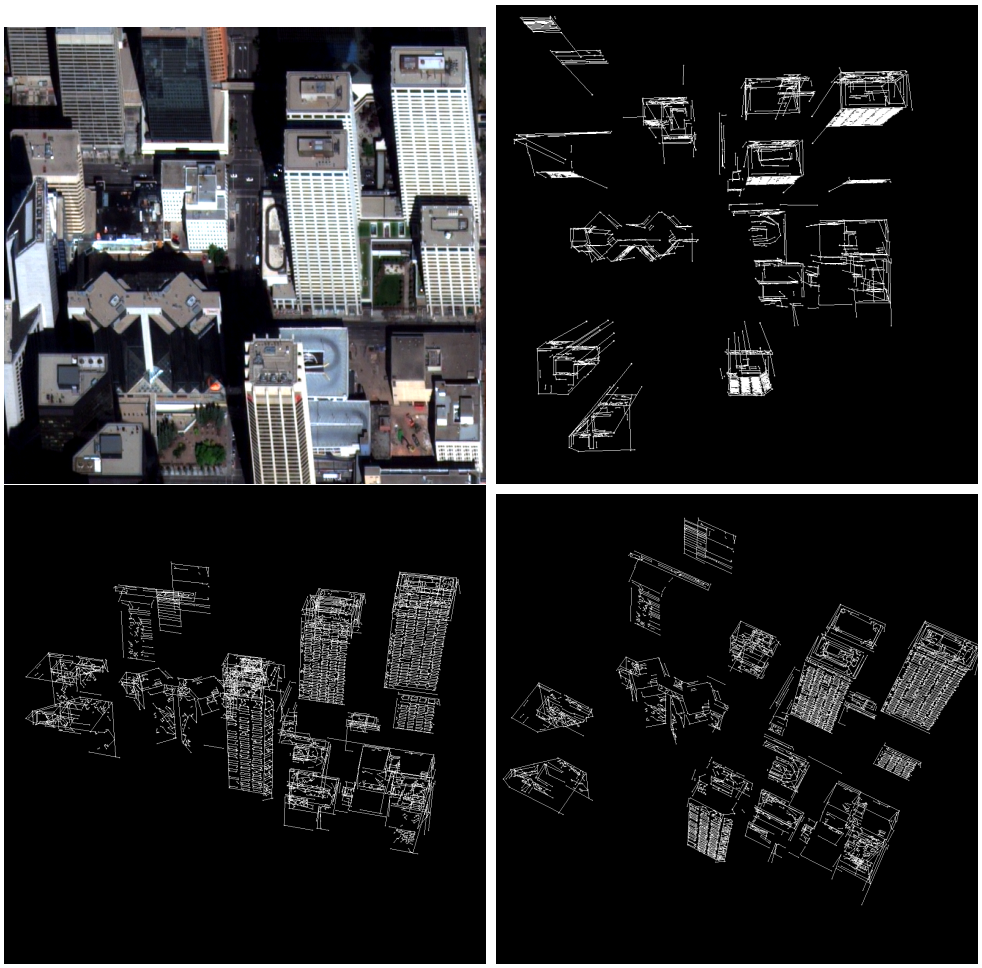


Figure 105: 3D Wireframe model of Calgary created from reconstructed boundaries by line matching. The top left image is on of the input image, top right shows the top view of the wireframe. The superstructures and parapet boundaries are visible. Bottom left shows the side view with some facade details reconstructed.

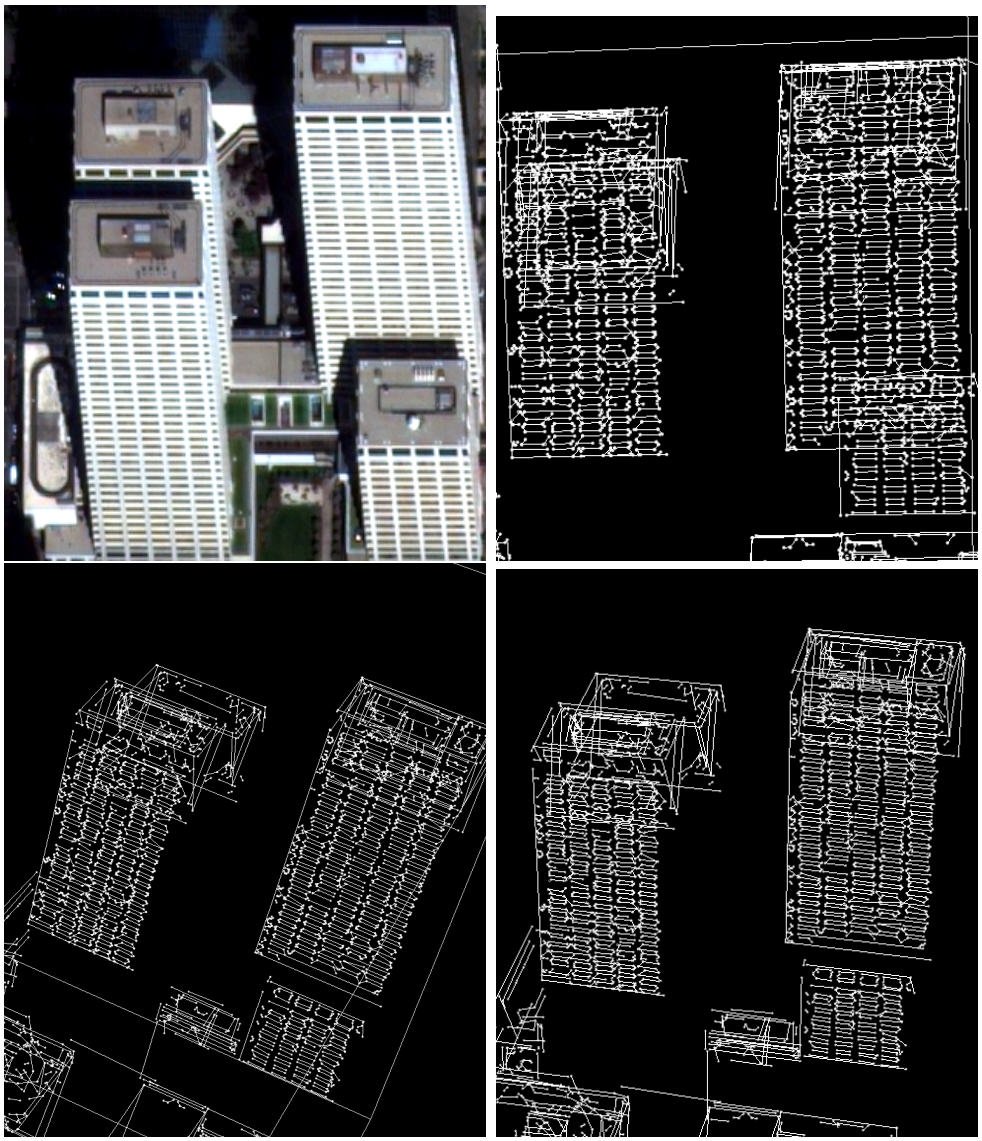


Figure 106: 3D Wireframe model of Calgary created from reconstructed boundaries by edge matching. The top left image is on of the input image, top right shows the top view of the wireframe. The superstructures and parapet boundaries are visible. Bottom left shows the side view with some facade details reconstructed.

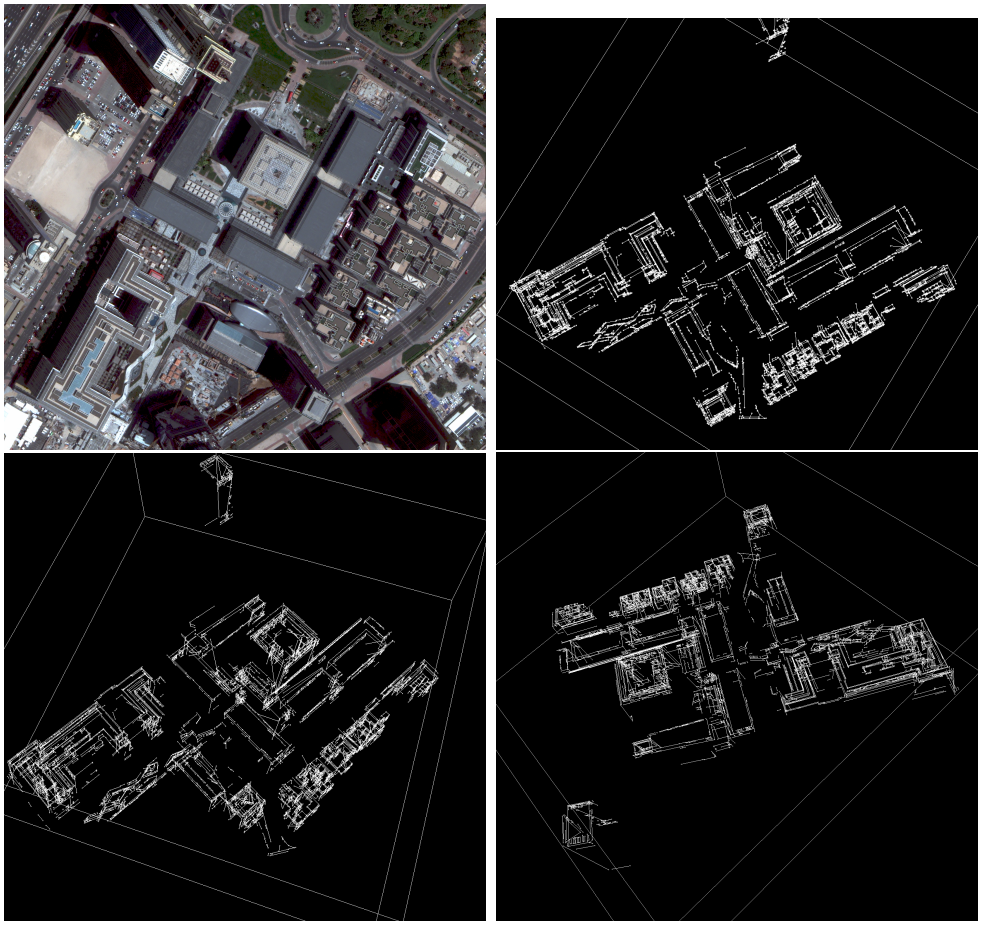


Figure 107: 3D Wireframe model of Dubai created from reconstructed boundaries by edge matching. The top left image is on of the input image, top right shows the top view of the wireframe. The superstructures and parapet boundaries are visible. Bottom left shows the side view with some facade details reconstructed.

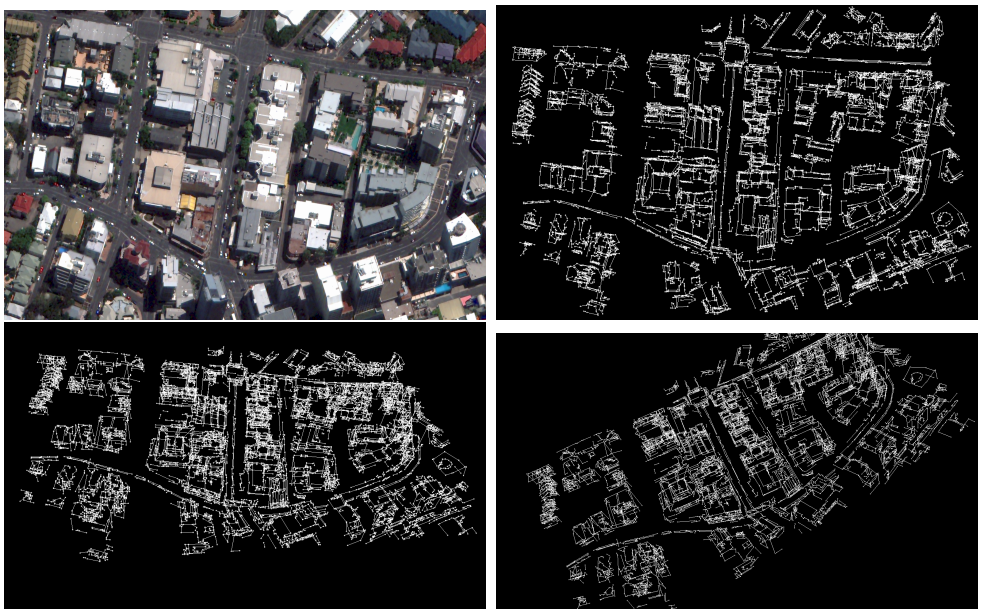


Figure 108: 3D Wireframe model of Brisbane created from reconstructed boundaries by edge matching. The top left image is on of the input image, top right shows the top view of the wireframe. The superstructures boundaries are visible.

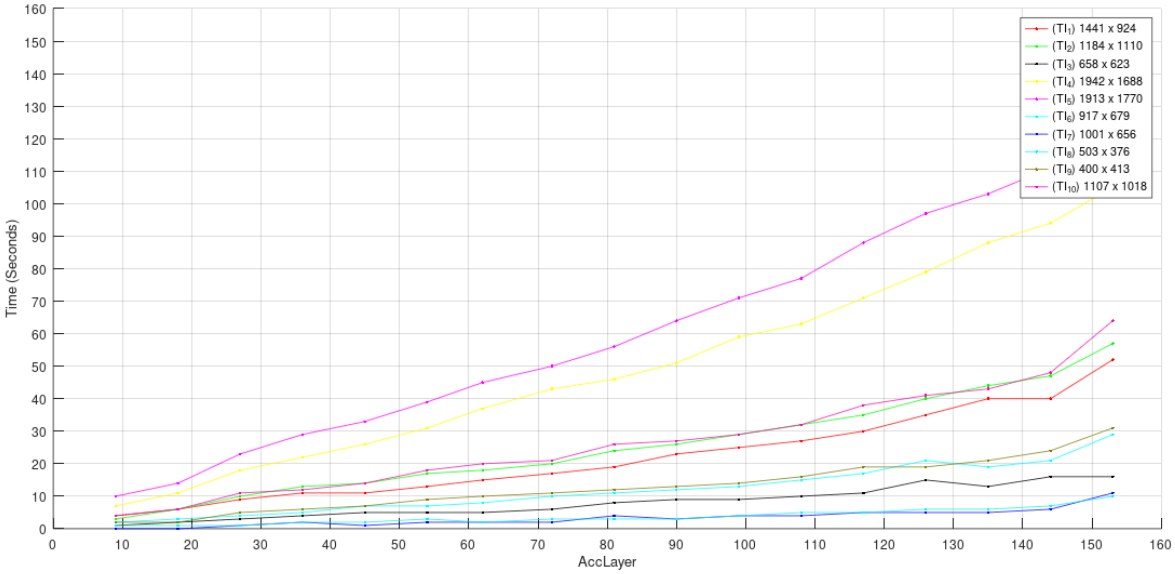


Figure 109: Time evaluation for ABED edge detection. The time taken during processing each accumulator level for each site is shown in the graph. There is a near-linear increase in processing time for subsequent levels.

From the results, ABED run-time was approximately 110s for the largest image of 1942 x 1688. The times are shorter for smaller images. For example,an image of 503 X 376 takes less than 10s to process. What is important to note is that run-time increases linearly as more accumulator levels are processed. The increase in time is thus near linear. The localisation and validation filters are thus not negatively affected by the number of edges nor the number of accumulator level. This implies that a tiling strategy for large images coupled with multi-core processing will further reduce the times recorded in this research. A tiling approach is recommended for large images, where each tile is potentially processed on its own core. The non-linear deviation in time at the tail end of each graph is due to rendering outputs for analysis, otherwise processing time increase can be considered linear.

Similarly, the edge detection and reconstruction processing times were plotted for the ten sites in Figure 110. In other words, the time required for the two combined frameworks was recorded. Similar behaviour is exhibited where the increase is almost linear. There is an increase in time with the largest image processed in approximately 240s or four minutes. If this image is tiled to the size of the smallest image shown in the results, which took approximately 20s, and processed in parallel, processing time can be drastically reduced.

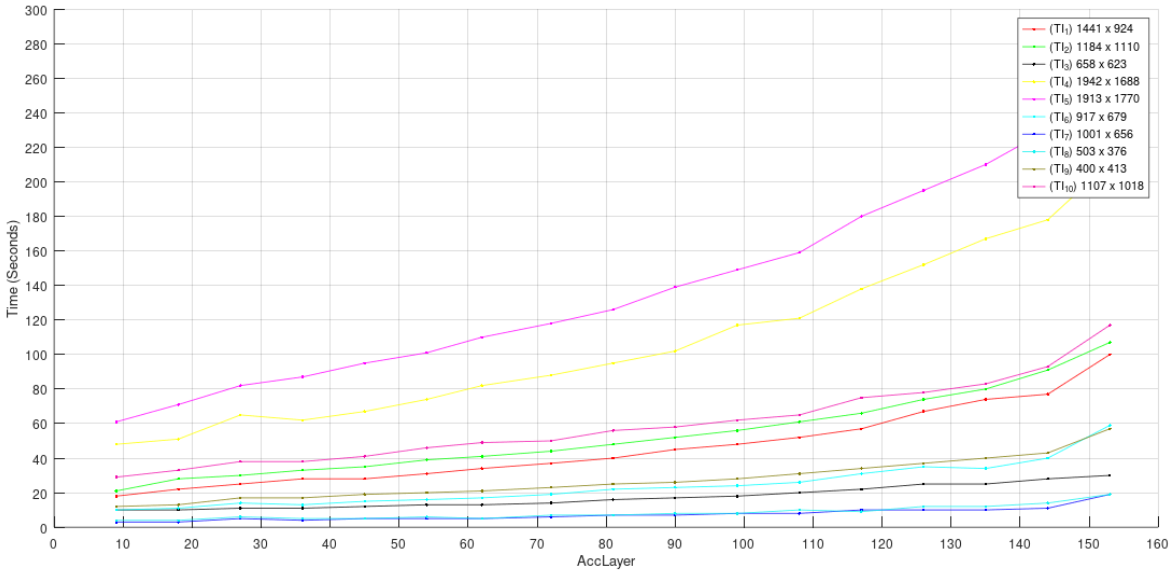


Figure 110: Time evaluation for ABED edge detection and ABEDL as a single process. The time taken during processing each accumulator level for all ten sites is shown in the graph. There is a near-linear increase in processing time with edge detection and reconstruction as a single process.

The performance of the framework was carried out on a Windows 10 machine with 64 Gig of Random access memory and an i7, six core processor.

7.7 Conclusion

This chapter presents the results of this research. Firstly, results of ABED framework were presented qualitatively and quantitatively for ten building sites. This was followed by the the qualitatively and quantitatively results of ABEDL on the ten sites used for ABED. The results on some recovered boundaries were shown together with some rendered 3D wire-frames. Performance results for ABED and ABEDL were also presented that show the overall processing times. Chapter 8 will give a detailed discussion of the results presented in this chapter, followed by conclusions and recommendations.

8 Discussion of Results

8.1 Introduction

This chapter discusses the results of the ABED and ABEDL frameworks. The research presented here aimed to solve pertinent issues associated with edge detection and building boundary shape reconstruction. Challenges in edge detection and boundary shape reconstruction were discussed in Chapter 2.5. Issues that affect the imaging process result in edge fragmentation and missing edges. The aim was to develop an edge detection and boundary shape recovery framework for satellite images that reduces fragmentation and missing edges, ultimately improving the resulting 2D and 3D building boundary shapes. For this, two sub-frameworks namely ABED and ABEDL were developed. In the development of these frameworks, two objectives were set out, namely:

1. Develop a novel generic edge detection framework that aggregates edges detected using multiple detectors with automated parameter tuning. The developed framework should recover edge features under varying imaging conditions from satellite images. The edges should represent both object form and detail. The edge detection framework must maintain good edge localisation.

2. Develop a boundary recovery framework that retrieves long unfragmented boundaries that model building boundary shapes from the edges detected by the first framework. Building boundaries are generally reconstructed using a set of lines. Consequently, the framework must detect and reconstruct connected lines that follow object boundaries in 2D and 3D. The recovered lines must result in detailed3D wireframes.

The two sub-frameworks were designed to improve the building boundary shapes in 2D and 3D. Accurate boundary shape estimation is achievable if maximal boundaries are retrieved minimising the effects of fragmented edge chains or rather discontinuous boundaries. Using stereo reconstruction and line matching techniques, detailed 3D models were produced. A discussion of the results for each objective is given in the next section.

8.2 Discussion

8.2.1 Objective 1

For the first objective, the Accumulator Based Edge Detector (ABED) framework was developed. The aim of the framework was to improve edge detection by reducing fragmentation and missing edges. This was achieved through edge aggregation using multiple detectors. Furthermore, the ABED framework performs automated parameter tuning during edge aggregation, eliminating the need to supply optimal edge detection parameters. Consequently, a gradient operator-agnostic framework was developed for edge detection. The framework was able to reduce fragmentation and missing edges across a different range of scenes. To measure performance, ten sites containing buildings were selected from satellite images. The ground truth edges were created and used for benchmarking with two parameter free edge detectors namely EDPF and CannyPF. The ten sites were chosen because of different architectural styles of buildings in each. These images had resolutions varying from $0.5m$ to $0.3m$.

From the results presented, ABED has shown the ability to recover more edges from satellite images, as evidenced by the improved recall rate in table 88. Figure 8 further illustrates the ability of the framework to grow edges. ABED, has shown, in addition, a strategy that can harness the strengths of various edge detectors and produce a homogeneous edge map that better captures object form and detail. By using an anchor map as a validation layer, weak edge pixels can be augmented to salient edge pixels with a good localisation strategy. The plot in Figure 87 shows similar localisation errors compared to state of the art detectors namely EDPF and CannyPF. To further illustrate the performance of the edge detection framework, Figures 84 and 85 show the edges recovered in dark and shadow regions in an image.

Most edge detectors are based on either the magnitude or direction of the gradient. Their performance relies on the identification of pixels where sharp intensity changes occur. To get the final edge maps, thresholding is required to filter out non-edge pixels. Selecting the threshold parameters is scene-specific. Thus, there is no global parameter to use for thresholding due to variations in image properties. ABED framework solves this by defining a wide parameter space, where automated parameter variation is used to

recover edges. Figure 86 illustrates the performance of ABED under difficult conditions. Furthermore, ABED can be extended to use any binary edge map regardless of the edge detection method used. This opens up the possibility of using existing edge maps as controls and validation for edge maps produced. ABED uses an anchor validation method that improved edge detection recall rate for the ten sites as illustrated in Table 4.

Another challenge that exists with most edge detectors is the lack of a definition of what a relevant edge is. Irrelevant edges are usually defined based on a proposed task. For example, tree edges are considered noisy and irrelevant if the task is building detection. However, these edges are still valid, as they show the extent of other features in the scene. ABED attempts to make no distinction and recovers all edges, which might be regarded as noise by other edge detectors. However, the edge detector should not define the relevant edges. An edge detector must have the ability to recover all edges regardless of the task. Task-specific algorithms must then be developed to extract relevant edges from a dense edge map. As shown by the boundary reconstruction task, a dense edge map resulted in longer, unbroken lines as shown in Figures 95 and 96. However, considering that the ground truth images are not entirely accurate, a recall rate of above 0.7 shows that most object edges are captured. The results presented were limited to building boundary edges. Benchmarking with detailed ground truth object edges is thus required for further assessment. The major contribution of ABED, is the ability to use an operator-agnostic framework to enhance edge detection while avoiding parameter tuning.

Overall, ABED improves edge detection in difficult imaging conditions. Because ABED is operator-agnostic, more edges were retrieved. As illustrated in Table 5, and the plots in Figure 89, the edge detection rate improved with the addition of accumulator levels. To illustrate that ABED does not continuously add irrelevant edges, the growth rate for TI_3 and TI_7 is low. This implies that the accumulator only adds relevant edges that pass validation tests rather than continuous addition of edges. While adding edges, ABED maintained consistent and good localisation, comparable to state-of-the-art detectors. From the results shown, the first objective was achieved by the developed sub-framework. The dense edge map produced was used as input for the second objective of the study.

8.2.2 Objective 2

The second objective was to develop a building boundary shape reconstruction framework for buildings using satellite images that ultimately improves both 2D, and 3D boundaries in the case of multiview scenes. This was achieved using the ABEDL framework. The use of edge features for full boundary recovery has been limited due to edge fragmentation and missing edges. Since these issues were addressed by ABED framework, this provided an opportunity to develop a framework that can ingest a dense edge map and extract task-specific edges, which in this case were buildings. The ABEDL framework was developed which takes as input the dense edge map produced by ABED. The method works by first defining a scene as a series of connected line segments that form a closed network. Objects are generally represented as a series of piecewise connected linear segments, as explained in Section 5.1. From this definition, all objects boundaries can be defined by a series of connected line segments. By detecting connected line segments in a dense edge map, closed boundaries of objects are recovered. Therefore, the framework works by detecting linear segments and their connectivity to ensure that boundaries are reconstructed accurately. The performance of ABEDL was measured based on the quality and quantity of the detected line segments.

To evaluate the performance of the framework in line detection and reconstruction, an evaluation of each detected line versus the ground truth line was performed. Instead of assessing the overall line detection per scene, a one-to-one comparison was performed per line, and a line coverage measure or percentage length detected was computed. The detected percentage coverage depends on both the length and orientation of the lines. Thus, this measure reflects both the length and the accuracy of orientation of the detected line. Plotting the percentage coverage of each line shows the number and quality of the detected lines. The results are shown in Figures 95 and 96. ABEDL has more full-length or longer lines compared to eight state-of-the-art line detectors, namely EDLines, CannyLines, PPHT, LSD, AG3Lines, F-Clip, L-CNN and HT-HWAP. As shown in blue in the plots of Figures 95 and 96, ABEDL detects longer lines, and at least a detection is guaranteed for most of the line segments. The blue graphs indicate that more lines or part thereof were detected. The decay shape of the curves for ABEDL in Figures 95 and

96 is consistent in nine of the scenes in urban areas with the exception of the last sample TI_{10}, which is a residential area. As mentioned above, ABEDL does not recover short edges accurately. High levels of fragmentation are encountered. Since the area TI_{10} is residential, characterised by short edges of house boundaries and ovehanging trees, the graph of TI_{10} shows a sharper decay as a result. However, for large city blocks, the decay is similar. Looking at TI_7, deep learning based L-CNN performed better than EDLines, CannyLines, PPHT, LSD and AG3Lines with 243 lines out of 602 as shown in Table 6. This is because the scene was made of a single building, which resembles that deep learning training contexts, of large single buildings using terrestrial images.

Recall, precision, and F1-score were calculated for the ten sites and recorded in Table 7. Good F1-Scores with all detectors were recorded in ABEDL, showing improved recall. In addition to detecting most of the lines, ABEDL detected longer lines compared to ground truth. However, ground truth lines were limited to building boundaries. Consequently, the F1-Score could be improved if the ground truth contained all lines in the scene. ABEDL had the second best precision values. This is because ABEDL uses links to connect boundaries, and these links lowered the precision, since they are not considered valid lines. The use of links highlights the effectiveness of both ABED from the first objective, and ABEDL. ABEDL was also able to recover more full length lines compared to other line detectors. The effectiveness of any boundary recovery algorithm is the ability to recover full-length boundary components. ABEDL lays a strong foundation for more full-length and longer lines recovery.

Three of the tested methods were deep learning based methods namely F-Clip [7], HT-HWAP [9] and L-CNN [8]. These methods are currently ranked second, third, and fifth with regard to line detection and wireframe parsing on the Line Segment Detection on Wireframe Benchmarking Dataset [86]. From the assessment, the DL methods tested detected fewer lines compared to traditional line detectors.This is because deep learning methods tend to perform in environments that match the training data with regards to resolution and detail. However, the few lines detected by the methods have minimal or no fragmentation. This is evidenced by the number of full-length lines recovered in Figures 95 and 96. Since these methods were trained on ground based imagery, fewer lines were detected for most of the ten sites with the exception of TI_7, since it is composed of one building, which resembles the context in which deep models were trained.

The two sub-frameworks were used to recover building boundaries. ABEDL treats lines as a connected network of curves that form closed boundaries that can be recovered. Since the lines from ABED and ABEDL are not disjoint, the task of finding boundaries is solved by graph traversal methods. Furthermore, boundary completeness is enforced using Gestalt principles. Since we have a dense edge map, the task of finding closed boundaries was reduced to connecting lines and curves using links. By traversing a graph, links to boundaries were established, and closed boundaries formed, as illustrated in Figure 7.4. Figures 100, 120 and 121 show the detailed building boundaries for three of the sites in comparison. By parameter relaxation using the implemented BFS algorithm, boundaries are connected, enhanced, and aligned, producing closed structures. In the absence of links, Gestalt principles are effective in closing boundaries. The boundary detection results show the possibility of using the ABEDL with satellite images.

The advantages of using global connectivity information when recovering line features have been demonstrated as shown by the boundary recovery specific cases. Having information on the neighbouring lines and their junction points enhances the boundary reconstruction while providing constraints that limit boundary distortion. Even with improved boundary detection, closed boundary reconstruction was not achieved for all buildings. ABEDL still missed some linear segments, resulting in partial detections. However, the results show the potential of the ABEDL with further research. One contribution to be taken into account from the results is that line recovery is better treated as a network of connected segments, where global information is taken into account during line detection and recovery from images.

Finally, to ascertain the effectiveness of the overarching framework, the goal was to reconstruct 3D models that show improved wireframe of 3D buildings from satellite images. This goal was automatically achieved based on the completion of the first two objectives. Three of the multiview scenes were used to recover 3D model wireframes of buildings based on the recovered boundaries. This is shown in Figures 104, 105, 106, 107 and 108. The 3D wireframes show some building form and detail compared to what is recovered using pixel matching approaches, which are points in the 3D space. The recovered wireframe presents the possibility of dense edge matching as opposed to detailed pixel matching. This is one of the main contributions of the research. Small objects, specifically on rooftops, were not properly reconstructed, and these contribute

to the small kinks in the final boundaries.

8.3 Limitations

While a detection rate of approximately 80% was recorded, ABED does not perform well when recovering edges of small objects of length less than four pixels. Small objects are prominent in limited-resolution imaging sensors such as satellites. Smaller objects and corners often result in small kinks and spikes in the edge map. These spikes were illustrated in Figure 66, These topological inconsistencies in the edge maps are solved during boundary detection by generalisation or omission of kinks and spikes. However, ABED attempts to recover the distorted or generalised boundary shape of the object. Although distorted, these short segment boundaries are useful for boundary completion. Another property of ABED is the ability to create closed regions using direction-constrained snapping presented in Section 4.1.6.

Furthermore, ABED still requires some contrast variation to exist. As shown in Figure 85, while boundaries in the low contrast and dark regions have been recovered, they are still incomplete and fragmented. ABED, therefore fails in regions of low contrast. When trying to recover edges in the dark regions, noise begins to have a negative effect on the result, as shown in Figure 86. Some false positives are introduced on the roof of the building. However, edges that represent the building form are not compromised. Since some object detail is composed of short segments, their boundary shapes are not recovered accurately, this often results in noisy detail and the boundary shape of the detail is not correctly recovered.

The major drawback of Accumulator Based Edge Detector (ABED) is the run time. ABED processes each accumulator in near-linear time as shown in Figure 109. Although there are benefits to edge aggregation in an accumulator space, the run-time could prove prohibitive for real-time applications such as Simultaneous Localisation And Mapping (SLAM) and object tracking. An optimised strategy is required to improve run-time. Since ABED recovers all relevant edges, precision cannot be easily computed in the absence of matching ground-truth edges.

The limitations of recovering small edges in satellite images have been discussed. It

then follows that small line segments are detected as kinks and spikes. This has an impact on the resulting boundary shapes recovered. Figures 98 and 100 show small kinks that complete boundaries, but results in a boundary shape that is not clean. The task of boundary generalisation is vast on its own and constitutes another investigation. Consequently, this topic was not explored in depth in this investigation.

The recovered wireframes shown in section 7.5 did not result in solid CAD models. This is because links that connect boundaries where not considered during edge matching and incomplete boundaries still exist. Only lines of length greater that four pixels were considered during matching hence the resulting wireframes contain line fragments. This is an area that requires further improvements. The ability to close boundaries should be used to recover closed wireframe boundaries.

The height attribute is used to verify matching lines and consequently matching closed boundaries. This implies that a height attribute is used to detect false boundaries. This means that in the absence of heights, some closed boundaries which do not make up building shape boundaries will remain in the output. This will introduce noise to applications that require only buildings. This limits the use of the proposed method to using at least two multiview images. While building boundary detection is improved in shadow areas, false edges from shadows remain in the edge map. This results in the reliance on more views to remove shadow effects.

Furthermore, occlusions and limited views result in the inability to find matches across edge maps. The number of facades that are visible in a given image limits the level of detail recovered in the final model and the density of the matching results. Given a single image, at most two building facades are visible if the facade normal and the satellite viewing direction do not lie in the xz plane. If the normals lie in the xz plane, then one facade is visible for the building. This limits the detail recovered around the buildings. This limitation can be overcome by sourcing more views. However, with satellite data, more views are not readily available. Consequently, estimating missing facade details based on recovered visible detail is a possible solution to having limited views. With two or more views, any missing facade detail of the produced 3D wireframe can be augmented with facade detail recovered from visible facades. Assuming that building facades are made of identical detail, only a single facade detail is required to augment

missing facade details. By applying the recovered facade details to occluded facades using camera projection parameters, a detailed wireframe can be recovered. Deep Learning based methods can be leveraged to provide sufficient details and contextual information to enhance wireframe detail if terrestrial scenes of the same area are available.

Reconstruction is still limited to 2.5D and only visible vertical walls are reconstructed. Complex shapes (for example dome shaped buildings) that are on building walls are thus not preserved correctly [197, 199]. To give a visual summary on the perfomance and capabilities of ABED and ABEDL, Table 9 was created.

TASK	**ABED**	**ABEDL**
Automated Parameter tuning	Solved. This was achieved through the implementation of the accumulator	N/A
Edge Aggregation	Solved. This was achieved by the localisation filter, which ensured consisted edge localisation	N/A
Missing Edges	Partially solved. The main reason for missing edges is image resolution, which results in pixels mixing and blurred edges at object edges. The change in contrast in some areas is still limited in satellite images and the low gradient magnitude for the edges falls outside of the wide parameter space	N/A
Fragmentation	Edge fragmentation was partially solved. This is because some image areas do not have sufficient contrast to register a significant change in gradient magnitude in the image that fall within the wide parameter space. Additionally, salt and pepper effects in satellite imagery are not entirely eliminated by image smoothing	N/A
Edges in Low Contrast Regions	Partially solved since the requirement is that the gradient magnitude for an edge pixel must line within the wide parameter space. if the image areas do not have sufficient contrast to return a gradient magnitude value that is within the wide parameter space, that edge is not detected	N/A
Edges in Shadow and Dark Regions	Partially solved since shadows suppress contrast. If the image areas under shadows do not have sufficient contrast to return a gradient magnitude value that is within the wide parameter space, that edge is not detected	N/A

TASK	ABED	ABEDL
Line and Curve Detection	N/A	Solved. If the edge was detected, then the lines and curves are detected.
Line Reconstruction	N/A	Partially Solved. This is because edges with a length of three pixel are not used to recover lines since three pixels are not sufficient to reduce uncertainty in line fitting. This causes fragmentation on the lines.
Curve Reconstruction	N/A	Solved. However, this process is entirely dependent on the edge map produced. If the curve edges are captured in the edge map, then a detection is always guaranteed.
Closed Boundary Reconstruction	N/A	Partially solved. Since fragmentation still exists and missing edges, full reconstruction is only achieved were there is no fragmentation and missing edges.
3D wireframe reconstruction with detail	N/A	Partially Solved. 3D wireframes were constructed with results from ABEDL. However, the limited views and occlusions of satellite images in addition to the existing fragmentation, resulted in missing detail.
Solid BREP/CAD Models	N/A	Not solved. Since there is fragmentation and missing edges, in addition to challenges in edge matching of short segments, solid BREP models are not produced. Furthermore, the limited views results in partial facade detail missing for occluded regions.

Table 9: Summary of the capabilities of the presented frameworks ABED and ABEDL in resolving existing challenges in 3D wireframe reconstruction from satellite images. The table indicates the challenges that have been solved and the extent of the solution.

8.4 Conclusion

Although all objectives were achieved, various areas that require more attention and improvement arose from the research. While there was a significant increase in recovered edges, and ultimately boundary shapes, not all lines were fully recovered. Partial detection still implies missing edge pixels. This could be attributed to some edge not passing the validation length and the minimum acceptable edge length. Furthermore, an optimised approach to edge accumulation is required to increase run-time efficiency. Multi-core and distributed processing techniques can be explored. The limited level of detail with reduced resolution sensors results in inaccurate modelling of small objects, even though their presence is detected. Research into

boundary generalisation methods with minimal shape distortion is required. Finally, the dense edge map presents a new challenge for edge-matching algorithms. The increased boundaries recovered open possibilities to better improve edge matching techniques. The next chapter discusses conclusions drawn and recommendations for future work.

9 Conclusions and Recommendations

This research has presented an edge-based building boundary reconstruction framework composed of two subframeworks namely ABED framework and the ABEDL framework. In combination, the frameworks improve building boundary shapes by improving the edge and boundary detection process. Traditionally, the use of edges as the main feature for reconstruction has been limited by the inability to recover longer unfragmented edges from images. This research has shown that with sufficient edge density, edge features can be used for 3D building boundary shape reconstruction. The challenges previously faced with shape recovery have been the lack of a boundary retrieval algorithms that detects closed building boundaries under different imaging conditions. Edge fragmentation and missing edges hinder the successful detection of maximal boundaries, hence attempts to recover shape form these boundaries have not been overly successful. Furthermore, the need for detection thresholds for most gradient-based edge detectors introduces further complexities in retrieving maximal boundaries.

The research presented here solves some of these edge detection challenges by introducing a parameter free edge detection framework, ABED, that is gradient operator agnostic. The detection framework offers several advantages. Rather than trying to establish the best case specific detector, the framework works with any and multiple gradient based edge detectors as input. In addition, the framework allows for the incorporation of preprocessing tasks that can benefit from automated parameter tuning, for example, image smoothing. This introduces genericity while maximising the features retrieved. Multiple smoothing operators can be defined in the framework. This eliminates the need for trial and error to define optimal parameters. The same applies for the gradient operators, which require some thresholding values for filtering out non-edge pixels. This makes the framework parameter free, rather operating on a wide parameter space to establish good edges while maintaining good localisation. Good localisation is achieved through a localisation filter that ensures positional accuracy during edge growing while validating edges. The issues of a single edge with multiple locations resulting from aggregating results from different operators is eliminated. One major advantage is that the resulting edge map has enough edge density that allows for 3D reconstruction from linear features using satellite images. This provides an

alternative to dense pixel matching for 3D modelling. Another significant result of the ABED framework is the performance under low contrast and shadows where edges that are not visible are recovered. The framework showed the ability to recover edges under difficult conditions, highlighting the effectiveness of the automated parameter tuning process.

The end goal for most edge detection tasks is to retrieve the full object boundaries. Object boundaries consist of connected linear and curved segments. Most algorithms have attempted to retrieve only the linear parts of building object boundaries. Line detection on the resulting edge map is a requirement. However, most line detection algorithms suffer from a rotational error introduced by corners and junctions. This results in a directional error of the retrieved lines. End-point mismatches and fragmentation are evident as a result. ABEDL framework presented allows line detection with higher directional accuracy by maintaining edge connectivity. This allows for accurate object boundary retrieval. This implies higher accuracy for resulting object shapes in 3D reconstruction. Higher accuracy is achieved by treating the boundary recovery problem as a network of connected lines with orientation as a constraint. Line fitting is treated as a global problem, where neighbouring lines enforce constraints during line searching and fitting. Since the connectivity of the recovered lines is maintained, recovery of curved boundaries is possible. Curves are assumed to be a series of piecewise-linear segments with smoothly changing orientations. Thus, curves are recovered by identifying connected linear segments. The connected linear segments are not fragmented. The connected linear segments opens the possibility of shape-based image segmentation using the developed framework.

Recovery of closed building boundaries was achieved by graph traversal using the ABEDL framework. The benefits of a dense edge map is that, edge links are guaranteed to exist in the graph of edges. Boundary completion was achieved through traversal using parameter relaxation and thus finding the shortest edge links that connect boundaries. The links are usually edge segments of length less than four pixels. By establishing links boundaries are implicitly closed. Furthermore, Gestalt principles are used to complete boundaries where links do not exist. The number of detected lines increased compared to the state-of-the-art detectors under different illuminating conditions using ABEDL.

3D Reconstruction has been largely driven by point processes namely pixel based matching from stereo or multi-view images. The output is mostly a point cloud that is used in subsequent boundary and shape reconstruction. Edge features have been used mainly as support in providing disambiguation in pixel matching strategies or augmenting features used in matching. This research has proven that, with enough edge density, full reconstruction is possible. Other features such as points can support edge reconstruction, with edges being the main feature. The reconstruction can be extended to other types of objects in addition to buildings. Therefore, this proves the hypotheses presented in Chapter 1. Images captured from space-borne platform all have projective distortion and are rarely orthophotos. Consequently, the retrieved boundaries are in projected image space. To retrieve, the true 3D positions of the boundary features, different image views are used to establish the edge correspondence and then retrieve 3D positions of matched features. This allows reconstruction of both the true 2D ad 3D positions of building structures. Recovered building wireframes in 3D can be used further to solve the true boundaries of the object by removing projective distortion.

ABED recovers a dense edge network that represents buildings and other image objects. Although the focus of the research was on building objects, the edges recovered by ABED can be used for other tasks. The edges recovered by ABED are classified according to the method explained in Section 7.3. These classified edges can be used to verify other image processing tasks, namely DSM generation, where breaklines are enhanced using edges or verify B-REP models created from other sources. This shows the unlimited application of enhanced edge detection presented in this research. Overall, the output from the proposed framework, allows the recovery of LOD 2 for buildings, according to OGC standards [94]. This opens the possibility of higher-level detailed 3D reconstruction from satellite images, which is currently limited.

9.1 Outcomes

The outcomes of the research are as follows:

9.1.1 Accumulator based edge detection framework (ABED)

An edge detection framework that recovers a dense edge map representing both object form and details and operates on a wide parameter space was developed. The edge detection framework termed ABED has various advantages that improve edge detection namely:

- The ABED framework is gradient operator agnostic. This allows the use of multiple detectors to leverage the different strengths that each of them individually present. The ABED framework can be extended to incorporate other functions that operate with predefined parameters. The need for experimentation in finding optimal parameters is removed. Automated parameter variation on a wide parameter space allows for the recovery of both salient and weak edges.

- The framework provides a localisation filter that ensures good edge localisation (Edge Location) and topological consistency of detected edges. A strategy to validate, merge and aggregate was presented. This provides a solution to vector fusion strategies and can be implemented for any other vector fusion exercises.

- Longer unbroken edge chains were recovered from satellite images under varying imaging conditions including dark and shadowy areas without the need for parameter tuning. Longer edges benefit various computer vision task with regards to low resolution imagery.

- Another major conclusion is that edge detectors should not be task specific. Edge detectors should aim to recover every edge in an image regardless of edge strength. The application that requires the edges should define a set of properties and algorithms for selecting relevant edges from the edge maps, as opposed to the edge detector returning the edges required by the application. This will avoid the need for parameter selection.

9.1.2 Building Boundary shape Recovery Framework (ABEDL)

An edge map contains both object form and detail. From the edges, line and curve recovery is required to reconstruct building boundary shapes. A framework for

recovering line and curves from dense edge map was developed termed ABEDL framework. By recovering a network of piecewise connected linear segments, boundary shapes are recovered. The results from the framework show vast improvements over existing line recovery methods in literature. The main contributions of the ABEDL framework are:

- Most line recovery methods in literature require a minimisation function that detects lines locally from edge pixels. Lines are treated as independent entities. This assumption often leads to over fitting of lines, line orientation biases, missed and fragmented lines. In addition, line end points rarely intersect. However, the framework presented here solves these issues by presenting lines as network of connected linear segments in a graph. During line search and fitting, global line information is used as constraints. Adjacent lines introduce constraints that preserve orientation and connectivity when a new line is fit. This ultimately results in reduced fragmentation and improved line orientation. Line search and recovery in images should be treated as a global problem as opposed to a local problem.

- Shorter lines are usually not detected, since minimisation approaches require a minimum number of points for a reliable fit. Consequently, shorter lines are eliminated by most line recovery methods. However, by using the line classification approach, short edges are retained as links and are used for completing boundaries,thereby reducing fragmented boundaries. Retaining shorter lines allows one to retain object detail, such as windows and chimneys.

- The line classification approach presented avoids the need for line fitting by simply connecting lineset end-points as an option. In cases where real-time line estimates are required, this is a desirable feature. An additional and optional line fitting refinement approach that improves line orientation was presented for any line recovery algorithm. Line orientation errors arise where there is high edge fragmentation or along short building edges. The line fitting refinement presented solves for this by correcting line orientation by an iterative fitting that starts from point set centre, radiating outwards. This eliminates outliers at line end points, which are usually responsible for distorting line orientations.

- The task of curve recovery is simplified by the ability to detect most lines from a dense edge map. Piecewise connected linear segments are recovered by simply traversing a graph.

9.1.3 Detailed 3D Wireframes

Shape recovery has been largely limited to rectilinear shapes or primitive libraries that fit an existing 3D point cloud or DSM. This has limited the shape recovered to building footprints for most algorithms. The shape recovery presented allows for the modelling of object detail, namely windows and superstructures using edge detection. This ultimately produces a more As-built representation of objects in 3D.

- The main contribution here is that detailed 3D wireframes of building models are achievable using edge features as an alternative to pixel matching. The ability to produce a dense edge map paves the way for edge-based feature matching and reconstruction. The improved line and curve recovery reduces matching ambiguities that are usually the limitation in line matching algorithms.
- Shape recovery can be reduced to a piecewise linear graph representation which simplifies the task of shape detection to a line graph segmentation task.

9.1.4 Additional Outcomes

Line simplification algorithm: One of the major challenges is edge-to-line simplification, which often leads to loss of accurate edge orientation. A line simplification approach based on an edge graph was developed through the BFS algorithm implemented. By constraining line fitting presented in section, shape generalisation is achieved without compromising line orientation by connecting lineset endpoints.

9.2 Recommendations

From the research presented here, recommendations and further research that can benefit and improve the findings are as follows:

- What was not investigated or established was the incorporation of image scaling in the edge accumulator. Image scaling could benefit detection by detecting parallel edges that belong to the same object. This is based on the assumption that by scaling an image, object form is retained, while object detail is lost. Reduced scale can be used to detect outer object boundaries, and full scale is used to capture object details. This could be of great benefit for edge matching and shape reconstruction.

- The research focuses on satellite image because it has the most in-scene variations for large areas. Furthermore, reduced resolution and pixel mixing further introduce challenges in detecting short edges. Since satellite image resolution is limited, smaller objects may be missed or incorrectly reconstructed. What would be of benefit is to assess performance on higher-resolution imagery with regard to edge detection.

- The reconstruction pipeline focused on building boundary shapes. However, dynamic urban environments are made up of other features, other than buildings. These features are much more complex in structure, and it would be beneficial to extend the research to detect other objects. Exploring the performance of the reconstruction pipeline on all objects is therefore required in a quest to achieve a full 3D scene reconstruction. The completion of wireframes using detected details is required. Often, satellite views do not cover all the facades of the buildings. An estimation step is required to reconstruct the details of the building on hidden facades.

- The focus was on detecting object form and detail for more accurate shapes to be reconstructed. However, a similar focus on edge matching procedures would greatly benefit the results. More research on how dense edges can be matched for reconstruction would thus be useful. Current matching algorithms are limited in that matching is currently done for line pairs. Consequently, links that close boundaries were dropped during matching.

- Performance and run-time optimisation of the accumulator is another factor that requires separate research. Since run-time and memory increase based on the size of the parameter space and operators used, it would be useful to research into optimisation techniques with regards to memory and run-time. While computing

resources are now available, efficiency is still critical in a world where near real time results are constantly required in some applications. Further to improving run-time is the investigation of dimension reduction of the accumulator layer count. A statistical analysis of relevant versus non-relevant layers would allow the reduction if the accumulator space, where only relevant accumulator layers are processed. Relevant layers are those that produce the most significant edges.

- The use of deep learning to recover wireframe detail using satellite imagery has no been tackled in this study. This is due to the lack of sufficient training data for satellite images for this study. The method proposed here provides a method that can be used in conjunction with deep learning methods to extract improved wireframes from images. For example, Luo et al. [83] uses LSD for generating lines as primary input to a deep learning algorithm that learns to build wireframes. Using edge or line segment output from the research presented here to enhance deep learning training and predictions similar to [83] is an example of how deep learning can be used with this research. Furthermore, from the results shown in Section 7.3.2, deep learning methods were able to recover some lines in all images, although less than half. This implies that sufficient training data on satellite imagery could potentially improve Deep Learning methods, and the results presented here can be used to generate such training samples and ground truth images for benchmarking.

Bibliography

[1] X. Lu, J. Yao, K. Li, and L. Li, "Cannylines: A parameter-free line segment detector," in *2015 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2015, pp. 507–511.

[2] C. Akinlar and C. Topal, "Edpf: A real-time parameter-free edge segment detector with a false detection control," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 26, no. 01, p. 1255002, 2012.

[3] ——, "Edlines: A real-time line segment detector with a false detection control," *Pattern Recognition Letters*, vol. 32, pp. 1633–1642, 2011.

[4] R. Grompone von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: a Line Segment Detector," *Image Processing On Line*, vol. 2, pp. 35–55, 2012, https://doi.org/10.5201/ipol.2012.gjmr-lsd.

[5] C. Galamhos, J. Matas, and J. Kittler, "Progressive probabilistic hough transform for line detection," in *Proceedings. 1999 IEEE computer society conference on computer vision and pattern recognition (Cat. No PR00149)*, vol. 1. IEEE, 1999, pp. 554–560.

[6] Y. Zhang, D. Wei, and Y. Li, "Ag3line: Active grouping and geometry-gradient combined validation for fast line segment extraction," *Pattern Recognition*, vol. 113, p. 107834, 2021.

[7] X. Dai, H. Gong, S. Wu, X. Yuan, and Y. Ma, "Fully convolutional line parsing," *Neurocomputing*, vol. 506, pp. 1–11, 2022.

[8] Y. Zhou, H. Qi, and Y. Ma, "End-to-end wireframe parsing," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 962–971.

[9] Y. Lin, S. L. Pintea, and J. C. van Gemert, "Deep hough-transform line priors," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*. Springer, 2020, pp. 323–340.

[10] U. N. H. S. Programme, "World cities report 2020: The value of sustainable urbanization," 2020.

[11] C. Corbane, M. Pesaresi, T. Kemper, P. Politis, A. J. Florczyk, V. Syrris, M. Melchiorri, F. Sabo, and P. Soille, "Automated global delineation of human settlements from 40 years of landsat satellite data archives," *Big Earth Data*, vol. 3, no. 2, pp. 140–169, 2019.

[12] Y. Song, "Infrastructure and urban development: Evidence from chinese cities," in *Proc. 2012 L. Policy Conf*, 2012, pp. 21–60.

[13] G. Xu, T. Dong, P. B. Cobbinah, L. Jiao, N. S. Sumari, B. Chai, and Y. Liu, "Urban expansion and form changes across african cities with a global outlook: Spatiotemporal analysis of urban land densities," *Journal of Cleaner Production*, vol. 224, pp. 802–810, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0959652619309862

[14] M. Batty, K. W. Axhausen, F. Giannotti, A. Pozdnoukhov, A. Bazzani, M. Wachowicz, G. Ouzounis, and Y. Portugali, "Smart cities of the future," *The European Physical Journal Special Topics*, vol. 214, no. 1, pp. 481–518, 2012.

[15] R. Sánchez-Corcuera, A. Nuñez-Marcos, J. Sesma-Solance, A. Bilbao-Jayo, R. Mulero, U. Zulaika, G. Azkune, and A. Almeida, "Smart cities survey: Technologies, application domains and challenges for the cities of the future," *International Journal of Distributed Sensor Networks*, vol. 15, no. 6, p. 1550147719853984, 2019.

[16] F. Biljecki, J. Stoter, H. Ledoux, S. Zlatanova, and A. Çöltekin, "Applications of 3d city models: State of the art review," *ISPRS International Journal of Geo-Information*, vol. 4, no. 4, pp. 2842–2889, 2015.

[17] U. D. Sulistyah and J.-H. Hong, "The use of 3d building data for disaster management: A 3d sdi perspective," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII-3/W8, pp. 395–402, 2019. [Online]. Available: https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-3-W8/395/2019/

[18] T. H. Kolbe, G. Gröger, and L. Plümer, "Citygml–3d city models and their potential for emergency response," in *Geospatial information technology for emergency response*. CRC Press, 2008, pp. 273–290.

[19] R. J. Lisle, "Google earth: a new geological resource," *Geology today*, vol. 22, no. 1, pp. 29–32, 2006.

[20] O. Mutanga and L. Kumar, "Google earth engine applications," 2019.

[21] Y. Chen, E. Shooraj, A. Rajabifard, and S. Sabri, "From ifc to 3d tiles: An integrated open-source solution for visualising bims on cesium," *ISPRS International Journal of Geo-Information*, vol. 7, no. 10, p. 393, 2018.

[22] J. E. Nichol, A. Shaker, and M.-S. Wong, "Application of high-resolution stereo satellite images to detailed landslide hazard assessment," *Geomorphology*, vol. 76, no. 1-2, pp. 68–75, 2006.

[23] N. L. Gavankar and S. K. Ghosh, "Automatic building footprint extraction from high-resolution satellite image using mathematical morphology," *European Journal of Remote Sensing*, vol. 51, no. 1, pp. 182–193, 2017.

[24] R. Qin, "Rpc stereo processor (rsp) â€“ a software package for digital surface model and orthophoto generation from satellite stereo imagery," in *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, ser. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, ISPRS, Ed., vol. III-1. Prague, Czech Republic: ISPRS, 2016, pp. 77–82.

[25] J. Heo, S. Jeong, H.-K. Park, J. Jung, S. Han, S. Hong, and H.-G. Sohn, "Productive high-complexity 3d city modeling with point clouds collected from terrestrial lidar," *Computers, Environment and Urban Systems*, vol. 41, pp. 26–38, 2013.

[26] B. Vajsova, A. Walczynska, S. Bärisch, P. J. Åstrand, and S. Hain, "New sensors benchmark report on worldview-4," *Publications Office Eur. Union, Luxembourg City, Luxembourg, Tech. Rep. EUR*, vol. 28761, 2017.

[27] DigitalGlobe. (2010) The benefits of the eight spectral bands of worldview-2. [Online]. Available: https://dg-cms-uploads-production.s3.amazonaws.com/uploads/document/file/35/DG-8SPECTRAL-WP_0.pdf

[28] ——. (2014) Accuracy of worldview products. [Online]. Available: https://dg-cms-uploads-production.s3.amazonaws.com/uploads/document/file/38/DG_ACCURACY_WP_V3.pdf

[29] Astrium. Spot 6 — spot 7 technical sheet. [Online]. Available: https://www.intelligence-airbusds.com/files/pmedia/edited/r18072_9_spot_6_technical_sheet.pdf

[30] ——. (2012) PlÃ©iades imagery user guide. [Online]. Available: http://satimagingcorp.s3.amazonaws.com/site/pdf/User_Guide_Pleiades.pdf

[31] DigitalGlobe. (2014) Worldview 3 datasheet.

[32] W. Sai, J. Ren, and Z. Jidong, "Superview-1-china's first commercial remote sensing satellite constellation with a high resolution of 0.5 m," *China Aerospace*, vol. 19, no. 1, pp. 31–38, 2019.

[33] M. Selva, L. Santurri, and S. Baronti, "Improving hypersharpening for worldview-3 data," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 6, pp. 987–991, 2018.

[34] N. Longbotham, F. Pacifici, S. Malitz, W. Baugh, and G. Camps-Valls, "Measuring the spatial and spectral performance of worldview-3," in *Hyperspectral Imaging and Sounding of the Environment*. Optica Publishing Group, 2015, pp. HW3B–2.

[35] DigitalGlobe. (2008) Geoeye-1. [Online]. Available: https://dg-cms-uploads-production.s3.amazonaws.com/uploads/document/file/97/DG_GeoEye1.pdf

[36] Y. Ma, H. Wu, L. Wang, B. Huang, R. Ranjan, A. Zomaya, and W. Jie, "Remote sensing big data computing: Challenges and opportunities," *Future generation computer systems*, vol. 51, pp. 47–60, 2015.

[37] X. Yao, G. Li, J. Xia, J. Ben, Q. Cao, L. Zhao, Y. Ma, L. Zhang, and D. Zhu, "Enabling the big earth observation data via cloud computing and dggs:

Opportunities and challenges," *Remote sensing (Basel, Switzerland)*, vol. 12, no. 1, pp. 62–, 2019.

[38] C. Yang, Q. Huang, Z. Li, K. Liu, and F. Hu, "Big data and cloud computing: innovation opportunities and challenges," *International journal of digital earth*, vol. 10, no. 1, pp. 13–53, 2016.

[39] M. Chi, A. Plaza, J. A. Benediktsson, Z. Sun, J. Shen, and Y. Zhu, "Big data for remote sensing: Challenges and opportunities," *Proceedings of the IEEE*, vol. 104, no. 11, pp. 2207–2219, 2016.

[40] D. PoliEmail and I. Caravaggi, "3d modeling of large urban areas with stereo vhr satellite imagery: lessons learned," *Journal of the International Society for the Prevention and Mitigation of Natural Hazards*, vol. 68, pp. 53–78, 2013.

[41] A. P. McClune, P. E. Miller, J. P. Mills, and D. Holland, "Automatic urban 3d building reconstruction from multi-ray photogrammetry," in *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-3. ISPRS, 2014, pp. 219–226.

[42] R. Wang, J. Peethambaran, and D. Chen, "Lidar point clouds to 3d urban models:a review," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, pp. 606 – 627, 2018.

[43] T. Hermosilla, L. A. Ruiz, J. A. Recio, and J. Estornell, "Evaluation of automatic building detection approaches combining high resolution images and lidar data," *Remote Sensing*, vol. 3, pp. 1188–1210, 2011.

[44] A. Wehr and U. Lohr, "Airborne laser scanning—an introduction and overview," *ISPRS Journal of photogrammetry and remote sensing*, vol. 54, no. 2-3, pp. 68–82, 1999.

[45] N. Pfeifer and C. Briese, "Geometrical aspects of airborne laser scanning and terrestrial laser scanning," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 36, no. 3/W52, pp. 311–319, 2007.

[46] S. Tripodi, N. Girard, G. Fonteix, L. Duan, W. Mapurisa, M. Leras, F. Trastour, Y. Tarabalka, and L. Laurore, "Brightearth: Pipeline for on-the-fly 3d

reconstruction of urban and rural scenes from one satellite image," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 3, pp. 263–270, 2022.

[47] N. Haala and M. Kada, "An update on automatic 3d building reconstruction," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 65, pp. 570–580, 2010.

[48] L. Zhang and A. Gruen, "Multi-image matching for dsm generation from ikonos imagery," *ISPRS Journal of Photogrammetry & Remote Sensing*, vol. 60, pp. 195–211, 2006.

[49] K. Gong and D. Fritsch, "Point cloud and digital surface model generation from high resolution multiple view stereo satellite imagery," in *The International Archives of the Photogrammetry, Remote Sensing and Spatial information Sciences*, ISPRS, Ed., vol. XLII-2. Riva del Garda, Italy: ISPRS TC II Mid-term Symposium â€œTowards Photogrammetry 2020, 2018, pp. 363–370.

[50] G. Facciolo, C. de Franchis, and E. Meinhardt-Llopis, "Automatic 3d reconstruction from multi-date satellite images," in *IEEE International Conference on Computer Vision and Pattern Recognition, EARTHVISION Workshop*, 2017.

[51] T. KrauÃŸ, M. Lehner, and P. Reinartz, "Generation of coarse 3d models of urban areas from high resolution stereo satellite images," in *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, ISPRS, Ed., vol. XXXVII, 2008.

[52] C. S. Fraser and H. B. Hanley, "Bias-compensated rpcs for sensor orientation of high-resolution satellite imagery," *Photogrammetric Engineering & Remote Sensing*, vol. 71, no. 8, pp. 900–915, 2005.

[53] G. Gupta, M. Rawat, R. Balasubramanian, R. Bhargava, and B. Krishna, "Region growing stereo matching method for 3d building reconstruction," *International Journal of Computational Vision and Robotics*, vol. 2, pp. 89–98, 2011.

[54] D. Weinshall and A. Golbert, "Object detection in multi-view 3d reconstruction using semantic and geometric context," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. II-3/W3, pp. 97–102, 2013.

[55] Y. Hu, X. Hu, P. Li, and Y. Ding, "Building detection from orthophotos using binary feature classification," *Multimedia Tools Appl.*, vol. 77, no. 3, pp. 3339–3351, 2018.

[56] G. Facciolo, C. de Franchis, and E. Meinhardt, "Mgm: A significantly more global matching for stereovision," in *Proceedings of the British Machine Vision Conference (BMVC*, B. Press, Ed. BMVA Press, 2016, p. 90.1â€"90.12.

[57] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328 – 341, 2008.

[58] F. Nex and F. Remondino, "Automatic roof outlines reconstruction from photogrammetric dsm," in *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, ISPRS, Ed., vol. I-3, Melbourne, Australia, 2012, pp. 257–262.

[59] E. Widyaningrum, R. Lindenbergh, B. Gorte, and K. Zhou, "Extraction of building roof edges from lidar data to optimize the digital surface model for true orthophoto generation," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, vol. XLII-2, pp. 1199–1204, 2018.

[60] N. Demir and E. Baltsavias, "Automated modeling of 3d building roofs using image and lidar data," in *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. I-4. Melbourne, Australia: ISPRS, 2012.

[61] T. Partovi, H. Arefi, T. Krauss, and P. Reinartz, "Automatic model selection for 3d reconstruction of buildings from satellite imagery," in *SMPR 2013 Conference*, ser. ISPRS Archives, H. Arefi, M. A. Sharifi, P. Reinartz, and M. R. Delavar, Eds., vol. XL-1/W. ISPRS, 2013, pp. 315–320. [Online]. Available: https:/elib.dlr.de/84968/

[62] I. Ernst and H. Hirschmuller, "Mutual information based semi-global stereo matching on the gpu," in *4th International Symposium on Visual Computing (ISVC08)*, Las Vegas, NV, USA, 2008.

[63] P. de Angelo, "Improving semi-global matching: Cost aggregation and confidence measure," in *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, ser. XXIII ISPRS Congress, ISPRS, Ed., vol. XLI-B1, Prague, Czech Republic, 2016.

[64] P. Reinartz, J. Tian, H. Arefi, T. Krauss, G. Kuschk, T. Partovi, and P. de Angelo, "Advances in dsm generation and higher level information extraction from high resolution optical stereo satellite data," in *Earsel Special Interest Group â€œ3D Remote Sensing & Urban Remote Sensing*, 2014.

[65] J.-P. Bauchet, W. Mapurisa, A. Gobbin, S. Tripodi, Y. Tarabalka, L. Duan, and L. Laurore, "Rooftops or footprints? reliable building footprint extraction from high-resolution satellite images," in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*. IEEE, 2021, pp. 274–277.

[66] N. Girard, D. Smirnov, J. Solomon, and Y. Tarabalka, "Polygonal building extraction by frame field learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5891–5900.

[67] X. Qin, S. He, X. Yang, M. Dehghan, Q. Qin, and J. Martin, "Accurate outline extraction of individual building from very high-resolution optical images," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 11, pp. 1775–1779, 2018.

[68] Z. Z. Jianghong Song Yanfeng Wei, "Urban building extraction from high-resolution satellite panchromatic image using clustering and edge detection," in *IGARSS 2004. 2004 IEEE International Geoscience and Remote Sensing Symposium*, vol. 3, 2004, pp. 2008–2010 vol.3.

[69] C. Topal and C. Akinlar, "Edge drawing: a combined real-time edge and segment detector," *Journal of Visual Communication and Image Representation*, vol. 23, no. 6, pp. 862–872, 2012.

[70] D. Ziou and S. Tabbone, "Edge detection techniques-an overview," *Pattern Recognition and Image Analysis C/C of Raspoznavaniye Obrazov I Analiz Izobrazhenii*, vol. 8, pp. 537–559, 1998.

[71] S. Savant, "A review on edge detection techniques for image segmentation," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 4, pp. 5898–5900, 2014.

[72] N. Ofir, M. Galun, S. Alpert, A. Brandt, B. Nadler, and R. Basri, "On detection of faint edges in noisy images," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 4, pp. 894–908, 2019.

[73] C. Zhang and C. Fraser, "Generation of digital surface model from high resolution satellite imagery," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci*, vol. 37, pp. 785–790, 2008.

[74] K. Li, J. Yao, X. Lu, L. Li, and Z. Zhang, "Hierarchical line matching based on line–junction–line structure descriptor and local homography estimation," *Neurocomputing*, vol. 184, pp. 207–220, 2016, roLoD: Robust Local Descriptors for Computer Vision 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231215018998

[75] S. Wang, F. Ge, and T. Liu, "Evaluating edge detection through boundary detection," *EURASIP Journal on Advances in Signal Processing*, vol. 2006, no. 1, p. 076278, 2006.

[76] B. Yüksel, Ç. Şenaras, M. Özay, and F. Y. Vural, "Automatic building detection from satellite images using stacked generalization architecture," in *2011 IEEE 19th Signal Processing and Communications Applications Conference (SIU)*. IEEE, 2011, pp. 694–697.

[77] A. Zakharov, A. Tuzhilkin, and A. Zhiznyakov, "Automatic building detection from satellite images using spectral graph theory," in *2015 International Conference on Mechanical Engineering, Automation and Control Systems (MEACS)*, 2015, pp. 1–5.

[78] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: A line segment detector," *Image Processing On Line*, vol. 2, pp. 35–55, 2012.

[79] Z. J. Liu, J. Wang, and W. P. Liu, "Building extraction from high resolution imagery based on multi-scale object oriented classification and probabilistic hough

transform," in *Proceedings. 2005 IEEE International Geoscience and Remote Sensing Symposium, 2005. IGARSS '05.*, vol. 4, 2005, pp. 2250–2253.

[80] C. Lin and R. Nevatia, "Building detection and description from a single intensity image," *Comput. Vis. Image Underst.*, vol. 72, no. 2, pp. 101–121, 1998. [Online]. Available: http://dx.doi.org/10.1006/cviu.1998.0724

[81] J. Xiaoying and D. C. H., "Automated building extraction from high-resolution satellite imagery in urban areas using structural, contextual, and spectral information," *EURASIP J. Appl. Signal Process.*, vol. 2005, pp. 2196–2206, 2005. [Online]. Available: http://dx.doi.org/10.1155/ASP.2005.2196

[82] X. Lin, Y. Zhou, Y. Liu, and C. Zhu, "A comprehensive review of image line segment detection and description: Taxonomies, comparisons, and challenges," *arXiv preprint arXiv:2305.00264*, 2023.

[83] Y. Luo, J. Ren, X. Zhe, D. Kang, Y. Xu, P. Wonka, and L. Bao, "Learning to construct 3d building wireframes from 3d line clouds," *arXiv preprint arXiv:2208.11948*, 2022.

[84] Y. Zhou, J. Huang, X. Dai, S. Liu, L. Luo, Z. Chen, and Y. Ma, "Holicity: A city-scale data platform for learning holistic 3d structures," *arXiv preprint arXiv:2008.03286*, 2020.

[85] Y. Zhou, H. Qi, Y. Zhai, Q. Sun, Z. Chen, L.-Y. Wei, and Y. Ma, "Learning to reconstruct 3d manhattan wireframes from a single image," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7698–7707.

[86] Y. U. Dataset. (2024) York urban dataset. [Online]. Available: https://paperswithcode.com/sota/line-segment-detection-on-wireframe-dataset

[87] ShanghaiTech. (2024) Shanghaitech urban dataset. [Online]. Available: https://paperswithcode.com/dataset/shanghaitech

[88] J. Huang, J. Stoter, R. Peters, and L. Nan, "City3d: Large-scale building reconstruction from airborne lidar point clouds," *Remote Sensing*, vol. 14, no. 9, p. 2254, 2022.

[89] H. Arefi and P. Reinartz, "Building reconstrcution using dsm and orthorectified images," *Remote Sensing*, vol. 5, no. 4, pp. 1681–1703, 2013.

[90] Y. Li and B. Wu, "Relation-constrained 3d reconstruction of buildings in metropolitan areas from photogrammetric point clouds," *Remote Sensing*, vol. 13, no. 1, p. 129, 2021.

[91] T. H. Kolbe, T. Kutzner, C. S. Smyth, C. Nagel, C. Roensdorf, and C. Heazel, "Ogc city geography markup language (citygml) part 1: Conceptual model standard," 2021, [Online; posted 13-September-2021]. [Online]. Available: https://docs.ogc.org/is/20-010/20-010.html

[92] L. Duan and F. Lafarge, "Towards large-scale city reconstruction from satellites," in *European Conference on Computer Vision*. Springer, 2016, pp. 89–104.

[93] Y. Wang, S. Schultz, and F. Giuffrida, "Pictometry's proprietary airborne digital imaging system and its application in 3d city modelling," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 37, no. B1, pp. 1–065, 2008.

[94] F. Biljecki, H. Ledoux, and J. Stoter, "An improved lod specification for 3d building models," *Computers Environment and Urban Systems*, vol. 59, pp. 25–37, 2016.

[95] S. Kim, K. Lee, T. Hong, M. Kim, M. Jung, and Y. Song, "An integrated approach to realize multi-resolution of b-rep model," in *Proceedings of the 2005 ACM symposium on Solid and physical modeling*, 2005, pp. 153–162.

[96] M. Li, F. Lafarge, and R. Marlet, "Approximating shapes in images with low-complexity polygons," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8633–8641.

[97] M. Buyukdemircioglu, R. Can, S. Kocaman, and M. Kada, "Deep learning based building footprint extraction from very high resolution true orthophotos and ndsm," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 2, pp. 211–218, 2022.

[98] W. Mapurisa and G. Sithole, "Improved edge detection for satellite images," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 2, pp. 185–192, 2022.

[99] J. Li and J. Cao, "A framework for automatic building detection from low-contrast vhr satellite imagery," in *Proceedings of the 3rd International Conference on Video and Image Processing*, 2019, pp. 52–56.

[100] D. Grigillo and U. Kanjir, "Urban object extraction from digital surface model and digital aerial images," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. I-3, pp. 215–220, 2012.

[101] H. Hasegawa, K. Matsuo, M. Koarai, N. Watanabe, H. Masaharu, and Y. Fukushima, "Dem accuracy and the base to height (b/h) ratio of stereo images," *International Archives of Photogrammetry and Remote Sensing*, vol. 33, no. B4/1; PART 4, pp. 356–359, 2000.

[102] K. Jacobsen, "Dem generation from satellite data," *EARSeL Ghent*, vol. 273276, no. 4, 2003.

[103] Luxcarta. (2023) Luxcarta. [Online]. Available: https://www.luxcarta.com/company/

[104] J. Grodecki and G. Dial, "Block adjustment of high-resolution satellite images described by rational polynomials," *Photogrammetric Engineering & Remote Sensing*, vol. 69, no. 1, pp. 59–68, 2003.

[105] T. Toutin, "Error tracking in ikonos geometric processing using a 3d parametric model," *Photogrammetric Engineering & Remote Sensing*, vol. 69, no. 1, pp. 43–51, 2003.

[106] K. He, J. Sun, and X. Tang, "Guided image filtering," in *Computer Vision – ECCV 2010*, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 1–14.

[107] D. Chaudhuri, N. K. Kushwaha, A. Samal, and R. C. Agarwal, "Automatic building detection from high-resolution satellite images based on morphology and internal

gray variance," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 5, pp. 1767–1779, 2016.

[108] G. Dial and J. Grodecki, "Block adjustment with rational polynomial camera models," in *Proceedings of ASPRS 2002 Conference, Washington, DC*, 2002, pp. 22–26.

[109] T. Kim and Y. Han, "Integrated preprocessing of multitemporal very-high-resolution satellite images via conjugate points-based pseudo-invariant feature extraction," *Remote Sensing*, vol. 13, no. 19, 2021. [Online]. Available: https://www.mdpi.com/2072-4292/13/19/3990

[110] M. J. Canty and A. A. Nielsen, "Automatic radiometric normalization of multitemporal satellite imagery with the iteratively re-weighted mad transformation," *Remote sensing of environment*, vol. 112, no. 3, pp. 1025–1036, 2008.

[111] C. S. Fraser, G. Dial, and J. Grodecki, "Sensor orientation via rpcs," *ISPRS journal of Photogrammetry and Remote Sensing*, vol. 60, no. 3, pp. 182–194, 2006.

[112] M. Morgan, K. Kim, S. Jeong, and A. Habib, "Parallel projection modelling for linear array scanner scenes," in *XXth ISPRS Congress, Istanbul, Turkey, PS WG III/1: Sensor Pose Estimation*, 2004, pp. 52–57.

[113] T. Kim and I. Dowman, "Comparison of two physical sensor models for satellite images: position–rotation model and orbit–attitude model," *The Photogrammetric Record*, vol. 21, no. 114, pp. 110–123, 2006.

[114] Y. Hu, V. Tao, and A. Croitoru, "Understanding the rational function model: methods and applications," *International archives of photogrammetry and remote sensing*, vol. 20, no. 6, pp. 119–124, 2004.

[115] C. V. Tao and Y. Hu, "A comprehensive study of the rational function model for photogrammetric processing," *Photogrammetric engineering and remote sensing*, vol. 67, no. 12, pp. 1347–1358, 2001.

[116] X. Tong, S. Liu, and Q. Weng, "Bias-corrected rational polynomial coefficients for high accuracy geo-positioning of quickbird stereo imagery," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 65, no. 2, pp. 218–226, 2010.

[117] S. K. Singh, S. D. Naidu, T. Srinivasan, B. G. Krishna, and S. PK, "Rational polynomial modelling for cartosat-1 data," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 37, 2008.

[118] A. Habib, S. W. Shin, K. Kim, C. Kim, K.-I. Bang, E.-M. Kim, and D.-C. Lee, "Comprehensive analysis of sensor modeling alternatives for high resolution imaging satellites," *Photogrammetric Engineering and Remote Sensing*, vol. 73, no. 11, pp. 1241–1251, 2007.

[119] Y. Dong, W. Chen, H. Chang, Y. Zhang, R. Feng, and L. Meng, "Assessment of orthoimage and dem derived from zy-3 stereo image in northeastern china," *Survey Review*, vol. 48, no. 349, pp. 247–257, 2016.

[120] Z. Xiong and Y. Zhang, "A generic method for rpc refinement using ground control information," *Photogrammetric Engineering & Remote Sensing*, vol. 75, no. 9, pp. 1083–1092, 2009.

[121] W. Gan, H. Albanwan, and R. Qin, "Radiometric normalization of multitemporal landsat and sentinel-2 images using a reference modis product through spatiotemporal filtering," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 4000–4013, 2021.

[122] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.

[123] C. Tomasi, "Histograms of oriented gradients," *Computer Vision Sampler*, pp. 1–6, 2012.

[124] K. G. Derpanis, "The harris corner detector," *York University*, vol. 2, pp. 1–2, 2004.

[125] K. Li, J. Yao, M. Xia, and L. Li, "Joint point and line segment matching on wide-baseline stereo images," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2016, pp. 1–9.

[126] A. Ok, J. B, C. Heipke, F. Rottensteiner, U. Soergel, and V. Toprak, "A stereo line matching technique for aerial images based on a pair-wise relation approach," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, vol. 38, 2012.

[127] E. Baltsavias, Z. Li, and H. Eisenbeiss, "Dsm generation and interior orientation determination of ikonos images using a testfield in switzerland," *Photogrammetrie, Fernerkundung, Geoinformation*, vol. 1, pp. 41–54, 2006.

[128] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[129] G. Lowe, "Sift-the scale invariant feature transform," *Int. J*, vol. 2, no. 91-110, p. 2, 2004.

[130] Y. Chen and W. Yao, "Extraction of orthogonal building boundary from airborne lidar data based on feature dimension reduction," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 2, pp. 351–358, 2022.

[131] X. Yan and Y. Li, "A method of lane edge detection based on canny algorithm," in *2017 Chinese Automation Congress (CAC)*. IEEE, 2017, pp. 2120–2124.

[132] H. Hirschmuller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. San Diego, CA, USA: IEEE, 2005.

[133] J. Oh, "Novel approach to epipolar resampling of hrsi and satellite stereo imagery-based georeferencing of aerial images," Ph.D. dissertation, The Ohio State University, Ohio State, 2011.

[134] A. F. Habib, M. F. Morgan, S. Jeong, and K.-O. Kim, "Epipolar geometry of line cameras moving with constant velocity and attitude," *ETRI Journal*, vol. 27, no. 2, pp. 172–180, 2005.

[135] M. Morgan, "Epipolar resampling of linear array scanner scenes," Ph.D. dissertation, University of Calgary, 2004.

[136] T. Kim, "A study on the epipolarity of linear pushbroom images," *Photogrammetric Engineering & Remote Sensing*, vol. 66, no. 8, pp. 961–966, 2000.

[137] T. Krauss, P. Reinartz, M. Lehner, M. Schroeder, and Stilla, "Dem generation from very high resolution stereo satellite data in urban areas using dynamic programming," in *ISPRS Hannover Workshop*, 2005.

[138] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, pp. 7–42, 2002.

[139] M. Rothermel, K. Wenzel, D. Fritsch, and N. Haala, "Sure: Photogrammetric surface reconstruction from imagery," in *LC3D Workshop*, Berlin, Germany, 2012.

[140] F. Bethmann and T. Luhmann, "Semi-global matching in object space," in *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-3/W2. ISPRS, 2015.

[141] M. Buyukdemircioglu, R. Can, and S. Kocaman, "Deep learning based roof type classification using very high resolution aerial imagery," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 43, pp. 55–60, 2021.

[142] M. Roux and L. Dumas, "Building segmentation based on stereo information from satellite images," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 2, pp. 299–306, 2022.

[143] A. Saxena, M. Sun, and A. Y. Ng, "Learning 3-d scene structure from a single still image," in *2007 IEEE 11th International Conference on Computer Vision*. Rio de Janeiro, Brazil: IEEE, 2007.

[144] L. Abraham and M.Sasikumar, "Unsupervised building extraction from high resolution satellite images irrespective of rooftop structures," *International Journal of Image Processing (IJIP)*, vol. 6, no. 4, pp. 219–232, 2012.

[145] N. L. Gavankar and S. K. Ghosh, "Automatic building footprint extraction from high-resolution satellite image using mathematical morphology," *European Journal of Remote Sensing*, vol. 51, no. 1, pp. 182–193, 2018.

[146] R. Attarzadeh and M. Momeni, "Object-based building extraction from high resolution satellite imagery," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 39, no. B4, 2012.

[147] L. Yong and W. Huayi, "Adaptive building edge detection by combining lidar data and aerial images," in *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXVII, Beijing, 2008.

[148] H. Chen, M. Lin, H. Zhang, G. Yang, G.-S. Xia, X. Zheng, and L. Zhang, "Multi-level fusion of the multi-receptive fields contextual networks and disparity network for pairwise semantic stereo," in *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, 2019, pp. 4967–4970.

[149] Z. Li, Q. Xin, Y. Sun, and M. Cao, "A deep learning-based framework for automated extraction of building footprint polygons from very high-resolution aerial imagery," *Remote Sensing*, vol. 13, no. 18, p. 3630, 2021.

[150] B. Neupane, T. Horanont, and J. Aryal, "Deep learning-based semantic segmentation of urban features in satellite images: A review and meta-analysis," *Remote Sensing*, vol. 13, no. 4, p. 808, 2021.

[151] Microsoft. (2022) Microsoft Building Footprints information page. [Online]. Available: https://www.microsoft.com/en-us/maps/building-footprints

[152] ——. (2022) Microsoft Building Footprints Repository buildings repository. [Online]. Available: https://github.com/microsoft/GlobalMLBuildingFootprints

[153] J. Czakon, K. A. Kaczmarek, A. Pyskir, and P. Tarasiewicz, "Best practices for elegant experimentation in data science projects," *EuroPython*, vol. 6, no. 7, p. 18, 2018.

[154] Z. Li, J. D. Wegner, and A. Lucchi, "Topological map extraction from overhead images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1715–1724.

[155] Z. Li, H. Yan, T. Ai, and J. Chen, "Automated building generalization based on urban morphology and gestalt theory," *International Journal of Geographical Information Science*, vol. 18, no. 5, pp. 513–534, 2004.

[156] A. Desolneux, L. Moisan, and J.-M. Morel, "Meaningful alignments," *International journal of computer vision*, vol. 40, pp. 7–23, 2000.

[157] H. Hongyan, Z. Nan, and Y. Chunyu, "Pos-supported automatic digital surface model (dsm) generation," *The International Archives of the Photogrammetry*, vol. XL-1/W2, pp. 207,210, 2013. [Online]. Available: https:/doaj.org/article/c4589a5f03a14a1ab164b9005505e3cd

[158] A. Gruen and K. Wolff, "Dsm generation with alos/prism data using sat-pp," in *IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2007.

[159] R. G. Frederik Tack and G. Buyuksalih, "Semi-automatic city model extraction from tri-stereoscopic vhr satellite imagery," in *International Archives Of Photogrammetry, Remote Sensing And Spatial Information Sciences*, vol. 38, 2009, pp. 89–96.

[160] P. D'Angelo and G. Kuschk, "Dense multi-view stereo from satellite imagery," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2012, pp. 6944–6947.

[161] A. W Gruen, "Adaptive least squares correlation: A powerful image matching technique," *South African Journal of Photogrammetry, Remote Sensing and Cartography*, vol. 14, pp. 175–187, 1985.

[162] K. Wolff and A. Gruen, "Up to date dsm generation using high resolution satellite image data," in *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 37(B1). Beijing, China: ISPRS, 2008, pp. 1103 – 1108.

[163] M. Bosch, Z. Kurtz, S. Hagstrom, and M. Brown, "A multiple view stereo benchmark for satellite imagery," in *In Proceedings of the Applied Imagery Pattern Recognition Workshop (AIPR)*, Washington, DC, USA, 2016.

[164] J.-Y. Rau and L.-C. Chen, "Dsm generation from ikonos stereo imagery," *Journal of Photogrammetry and Remote Sensing*, vol. 10, no. 3, pp. 265–274, 2006.

[165] R. H. Alwan and M. A. Naji, "Automatic stereo image matching using edge detection technique," *International Archives of Photogrammetry and Remote Sensing*, vol. XXXI, pp. 29–35, 1996.

[166] G. Sithole and G. Vosselman, "Experimental comparison of filter algorithms for bare-earth extraction from airborne laser scanning point clouds," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 59, no. 1, pp. 85–101, 2004.

[167] Y. Liu, S. D'Aronco, K. Schindler, and J. D. Wegner, "Pc2wf: 3d wireframe reconstruction from raw point clouds," *arXiv preprint arXiv:2103.02766*, 2021.

[168] Y. Qian, H. Zhang, and Y. Furukawa, "Roof-gan: Learning to generate roof geometry and relations for residential houses," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2796–2805.

[169] W. Zhao, C. Persello, and A. Stein, "Building outline delineation: From aerial images to polygons with an improved end-to-end learning framework," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 175, pp. 119–131, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271621000551

[170] D. Griffiths and J. Boehm, "Improving public data for building segmentation from convolutional neural networks (cnns) for fused airborne lidar and image data using active contours," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 154, pp. 70–83, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271619301352

[171] C. Chawda, J. Aghav, and S. Udar, "Extracting building footprints from satellite images using convolutional neural networks," in *2018 International Conference on*

Advances in Computing, Communications and Informatics (ICACCI). IEEE, 2018, pp. 572–577.

[172] E. Chartock, W. LaRow, and V. Singh, "Extraction of building footprints from satellite imagery," *Stanford University Report*, 2017.

[173] D. Marmanis, K. Schindler, J. Wegner, S. Galliani, M. Datcu, and U. Stilla, "Classification with an edge: Improving semantic image segmentation with boundary detection," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 135, pp. 158–172, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S092427161630572X

[174] W. Sun and R. Wang, "Fully convolutional networks for semantic segmentation of very high resolution remotely sensed images combined with dsm," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 3, pp. 474–478, 2018.

[175] D. Yu, S. Ji, J. Liu, and S. Wei, "Automatic 3d building reconstruction from multi-view aerial images with deep learning," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 171, pp. 155–170, 2021.

[176] A. Henn, G. Gröger, V. Stroh, and L. Plümer, "Model driven reconstruction of roofs from sparse lidar point clouds," *ISPRS Journal of photogrammetry and remote sensing*, vol. 76, pp. 17–29, 2013.

[177] A. Dick, P. Torr, and R. Cipolla, "Modelling and interpretation of architecture from several images," *International Journal of Computer Vision*, vol. 60, pp. 111–134, 2004.

[178] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[179] B. Sirmacek and C. Unsalan, "A probabilistic framework to detect buildings in aerial and satellite images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 1, pp. 211–221, 2010.

[180] C. G. Harris and M. Stephens, "A combined corner and edge detector." in *Alvey vision conference*, vol. 15. Citeseer, 1988, pp. 10–5244.

[181] B. Sirmacek and C. Unsalan, "A probabilistic framework to detect buildings in aerial and satellite images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 1, pp. 211–221, 2011.

[182] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[183] I. Sobel and G. Feldman, "A 3x3 isotropic gradient operator for image processing, presented at a talk at the stanford artificial project," *Pattern Classification and Scene Analysis*, pp. 271–272, 3.

[184] M. Aamir, Y.-F. PU, Z. Rahman, M. Tahir, H. Naeem, and Q. Dai, "A framework for automatic building detection from low-contrast satellite images," *Symmetry*, vol. 11, p. 3, 2018.

[185] C. Adamos and W. Faig, "Hough transform in digital photogrammetry," *International Archives of Photogrammetry and Remote Sensing*, vol. 29, pp. 250–250, 1993.

[186] N. Jiang, J. Zhang, H. Li, and X. Lin, "Object-oriented buinding extraction by dsm and very high-resolution orthoimages," in *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXVII, Beijing, China, 2008, pp. 441–446.

[187] T. Partovi, H. Huang, T. KrauÃŸ, H. Mayer, and P. Reinartz, "Statistical building roof reconstruction from worldview-2 stereo imagery," in *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-3/W2. Munich, Germany: ISPRS, 2015.

[188] D. Poli and P. Soille, "Refinement of digital surface models through constrained connectivity partitioning of optical imagery," in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, ISPRS, Ed., vol. XXXVIII- XXXVIII-4/W19, ISPRS. Hannover, Germany: ISPRS, 2011, pp. 245–250.

[189] L. Nan and P. Wonka, "Polyfit: Polygonal surface reconstruction from point

clouds," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2353–2361.

[190] O. S. Map. (2014) Open street map. [Online]. Available: https://www.openstreetmap.org

[191] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: the international journal for geographic information and geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.

[192] S.-T. Wu and M. R. G. Marquez, "A non-self-intersection douglas-peucker algorithm," in *16th Brazilian symposium on computer graphics and Image Processing (SIBGRAPI 2003)*. IEEE, 2003, pp. 60–66.

[193] P. Schuegraf, J. Schnell, C. Henry, and K. Bittner, "Building section instance segmentation with combined classical and deep learning methods," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 2, pp. 407–414, 2022.

[194] L. Castrejon, K. Kundu, R. Urtasun, and S. Fidler, "Annotating object instances with a polygon-rnn," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5230–5238.

[195] D. Acuna, H. Ling, A. Kar, and S. Fidler, "Efficient interactive annotation of segmentation datasets with polygon-rnn++," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 859–868.

[196] H. Ling, J. Gao, A. Kar, W. Chen, and S. Fidler, "Fast interactive object annotation with curve-gcn," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5257–5266.

[197] E. Delage, H. Lee, and A. Ng, "A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, New York, NY, USA, 2006.

[198] A. Saxena, M. Sun, and A. Y. Ng, "Make3d: Learning 3d scene structure from a single still image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 824 – 840, 2009.

[199] F. Han and S.-C. Zhu, "Bayesian reconstruction of 3d shapes and scenes from a single image," in *First IEEE International Workshop on Higher-Level Knowledge in 3D Modeling and Motion Analysis*, IEEE. IEEE, 2003.

[200] H. Huang, C. Brenner, and M. Sester, "A generative statistical approach to automatic 3d building roof reconstruction from laser scanning data," *ISPRS Journal of photogrammetry and remote sensing*, vol. 79, pp. 29–43, 2013.

[201] R. Szeliski, *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[202] R. Sun, T. Lei, Q. Chen, Z. Wang, X. Du, W. Zhao, and A. K. Nandi, "Survey of image edge detection," *Frontiers in Signal Processing*, vol. 2, p. 826967, 2022.

[203] A. Goyal, A. Bijalwan, and M. K. Chowdhury, "A comprehensive review of image smoothing techniques," *International Journal of Advanced Research in Computer Engineering and Technology*, vol. 1, no. 4, pp. 315–319, 2012.

[204] M. Ali and D. Clausi, "Using the canny edge detector for feature extraction and enhancement of remote sensing images," in *IGARSS 2001. Scanning the Present and Resolving the Future. Proceedings. IEEE 2001 International Geoscience and Remote Sensing Symposium (Cat. No. 01CH37217)*, vol. 5. IEEE, 2001, pp. 2298–2300.

[205] H. Scharr, "Optimal operators in digital image processing dissertation," 2000.

[206] J. Prewitt, "Object enhancement and extraction picture processing and psychopictorics. 1970."

[207] L. Roberts, *Machine Perception of Three-dimensional Solids*, ser. Its Technical report. M.I.T. Lincoln Laboratory, 1963. [Online]. Available: https://books.google.co.za/books?id=HS97GwAACAAJ

[208] A. Baştürk and E. Günay, "Efficient edge detection in digital images using a cellular neural network optimized by differential evolution algorithm," *Expert Systems with applications*, vol. 36, no. 2, pp. 2645–2650, 2009.

[209] D. Argialas and O. Mavrantza, "Comparison of edge detection and hough transform techniques for the extraction of geologic features," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 34, no. Part XXX, 2004.

[210] A. Mittal, S. Sofat, E. Hancock, and S. Mousset, "A statistical operator for detecting weak edges in low contrast images," in *International Conference Image Analysis and Recognition*. Springer, 2012, pp. 89–96.

[211] G. Bertasius, J. Shi, and L. Torresani, "Deepedge: A multi-scale bifurcated deep network for top-down contour detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4380–4389.

[212] M. El-Sayed and Y. Estaitia, "Edge detection using convolutional neural network," *International Journal of Advanced Computer Science and Applications*, vol. 4, pp. 11–17, 2013.

[213] Y. Liu, M.-M. Cheng, X. Hu, K. Wang, and X. Bai, "Richer convolutional features for edge detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3000–3009.

[214] Y. Liu, M.-M. Cheng, D.-P. Fan, L. Zhang, J. Bian, and D. Tao, "Semantic edge detection with diverse deep supervision," *arXiv preprint arXiv:1804.02864*, 2018.

[215] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1395–1403.

[216] M. Pu, Y. Huang, Q. Guan, and H. Ling, "Rindnet: Edge detection for discontinuity in reflectance, illumination, normal and depth," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 6879–6888.

[217] M. A. El-Sayed, Y. A. Estaitia, and M. A. Khafagy, "Automated edge detection using convolutional neural network," *Int. J. Adv. Comput. Sci. Appl.(IJACSA)*, vol. 4, no. 10, 2013.

[218] S. M. Smith and J. M. Brady, "Susan: a new approach to low level image processing," *International journal of computer vision*, vol. 23, no. 1, pp. 45–78, 1997.

[219] D. H. Ballard, "Generalizing the hough transform to detect arbitrary shapes," *Pattern recognition*, vol. 13, no. 2, pp. 111–122, 1981.

[220] J. Matas, C. Galambos, and J. Kittler, "Robust detection of lines using the progressive probabilistic hough transform," *Computer vision and image understanding*, vol. 78, no. 1, pp. 119–137, 2000.

[221] L. A. Fernandes and M. M. Oliveira, "Real-time line detection through an improved hough transform voting scheme," *Pattern recognition*, vol. 41, no. 1, pp. 299–314, 2008.

[222] Y. Salaün, R. Marlet, and P. Monasse, "Multiscale line segment detector for robust and accurate sfm," in *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2016, pp. 2000–2005.

[223] N.-G. Cho, A. Yuille, and S.-W. Lee, "A novel linelet-based representation for line segment detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 5, pp. 1195–1208, 2017.

[224] W. K. Pratt, *Introduction to digital image processing*. CRC press, 2013.

[225] K. Huang, Y. Wang, Z. Zhou, T. Ding, S. Gao, and Y. Ma, "Learning to parse wireframes in images of man-made environments," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 626–635.

[226] S. Huang, F. Qin, P. Xiong, N. Ding, Y. He, and X. Liu, "Tp-lsd: Tri-points based line segment detector," in *European Conference on Computer Vision*. Springer, 2020, pp. 770–785.

[227] A. Ok, J. Wegner, C. Heipke, F. Rottensteiner, U. Soergel, and V. Toprak, "A new straight line reconstruction methodology from multi-spectral stereo aerial images," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, vol. 38, 2010.

[228] R. Jain, R. Kasturi, and B. G. Schunck, *Machine vision*. McGraw-Hill New York, 1995, vol. 5.

[229] R. Muthukrishnan and M. Radha, "Edge detection techniques for image segmentation," *International Journal of Computer Science & Information Technology*, vol. 3, no. 6, p. 259, 2011.

[230] P. Dhankhar and N. Sahu, "A review and research of edge detection techniques for image segmentation," *International Journal of Computer Science and Mobile Computing*, vol. 2, no. 7, pp. 86–92, 2013.

[231] A. Koschan, "A comparative study on color edge detection," in *Proceedings of the 2nd Asian Conference on Computer Vision*, vol. 3. ACCV Sigapore, Sigapore, 1995, pp. 574–578.

[232] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, "Edge-preserving decompositions for multi-scale tone and detail manipulation," *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2008)*, vol. 27, no. 3, p. to appear, 2008.

[233] S. Paris, P. Kornprobst, J. Tumblin, and F. Durand, "Bilateral filtering: Theory and applications," *Foundations and Trends® in Computer Graphics and Vision*, vol. 4, no. 1, pp. 1–73, 2009. [Online]. Available: http://dx.doi.org/10.1561/0600000020

[234] N. Tatar, H. Arefi, and M. Hahn, "High-resolution satellite stereo matching by object-based semiglobal matching and iterative guided edge-preserving filter," *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 10, pp. 1841–1845, 2021.

[235] S. shui Bao, Y. rui Huang, J. chang Xu, and G. yu Xu, "Pixel difference unmixing feature networks for edge detection," *IEEE Access*, 2023.

[236] H. Bay, V. Ferraris, and L. Van Gool, "Wide-baseline stereo matching with line segments," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, pp. 329–336 vol. 1.

[237] O. A. Pellejero, C. Sagüés, and J. J. Guerrero, "Automatic computation of the fundamental matrix from matched lines," in *Conference on Technology Transfer*. Springer, 2003, pp. 197–206.

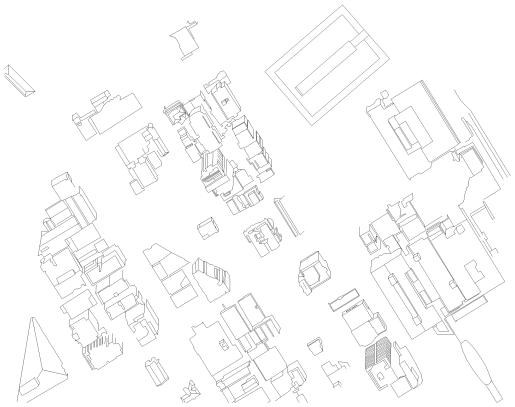
[238] C. Rother, "Linear multi-view reconstruction of points, lines, planes and cameras using a reference plane." in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2, 2003, pp. 1210–1217.

[239] O. Martorell, A. Buades, and B. Coll, "Matching of line segment for stereo computation." in *VISIGRAPP (6: VISAPP)*, 2017, pp. 410–417.

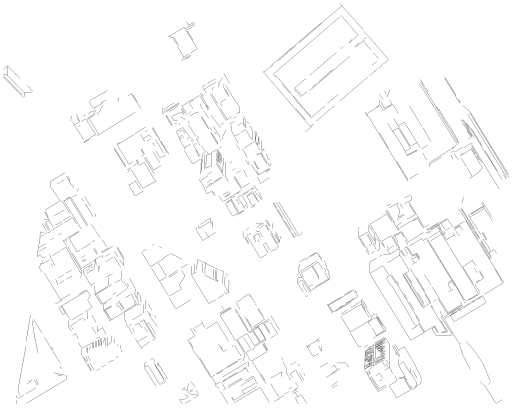
[240] B. G. Libraries. (2023) Boost graph libraries. [Online]. Available: https://www.boost.org/doc/libs/1_80_0/libs/graph/doc/index.html

10 Appendix A

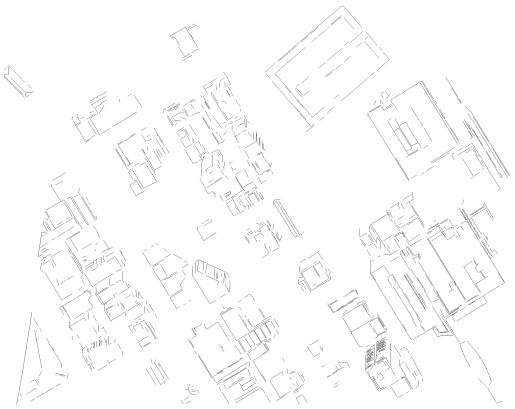
TI_4



GT

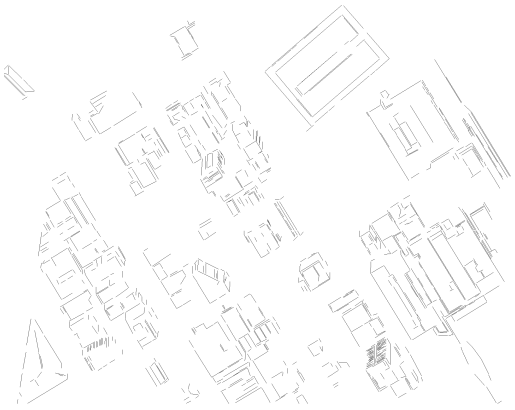


EDlines



LSDlines

PPHT



CannyLines

AG3Lines | ABEDL

Figure 111: Line detection results comparison for Brisbane CBD (TI_4)

TI_6 | GT

EDlines | LSDlines

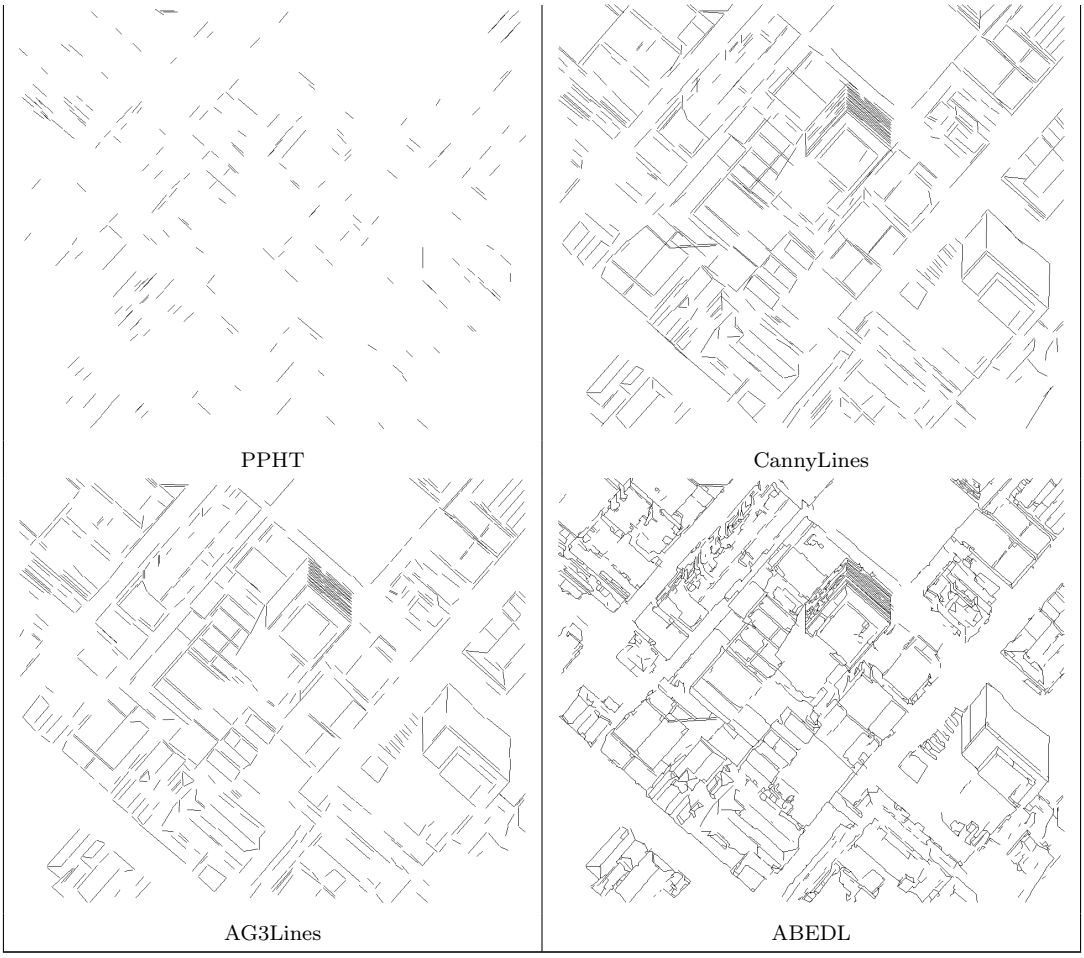
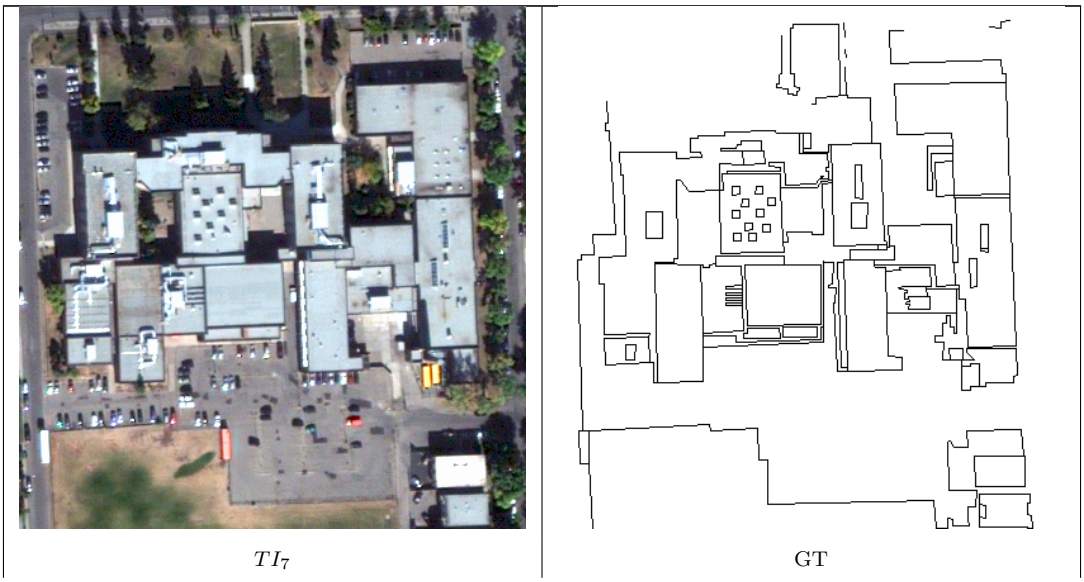


Figure 112: Line detection results comparison for Brisbane CBD (TI_6)



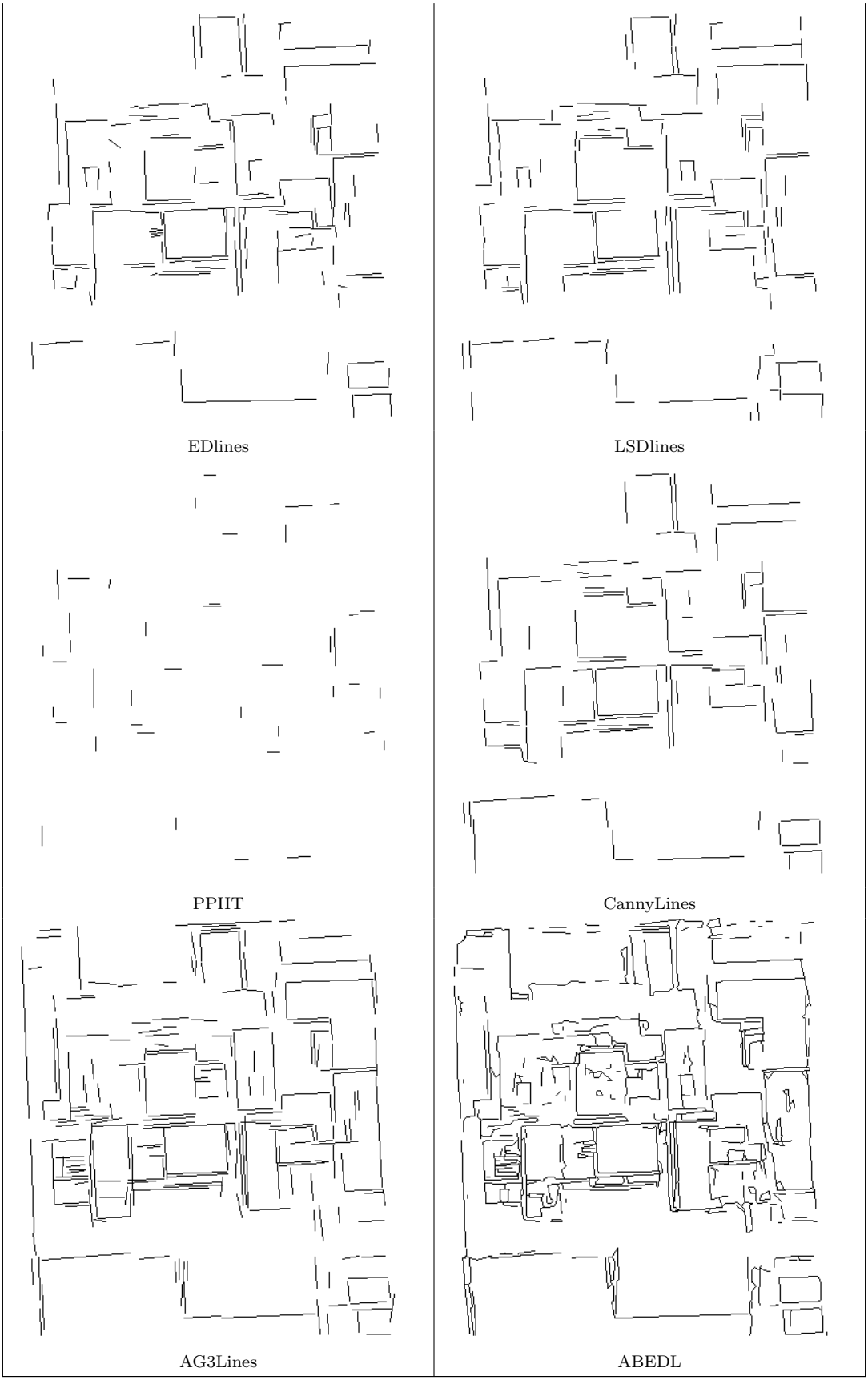


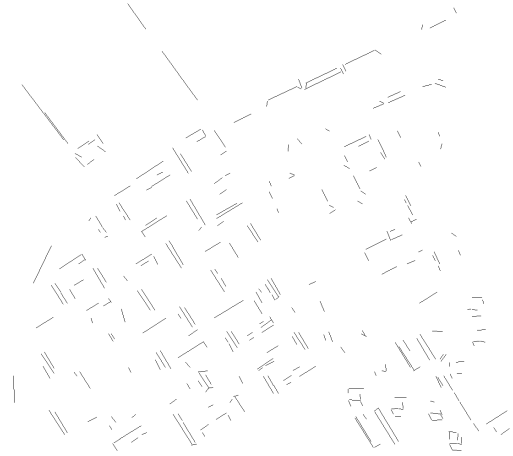
Figure 113: Line detection results comparison for an Industrial Building in Calgary(TI_7)

Figure 114: Line detection results comparison for a London residential area (TI_8)

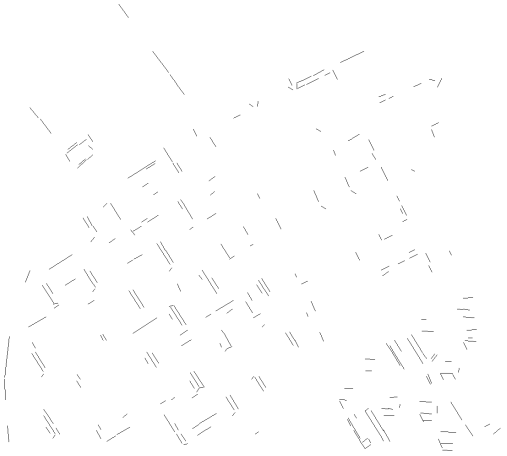
Figure 115: Line detection results comparison for Mourmelon CBD (TI_9)

TI_{10}

GT

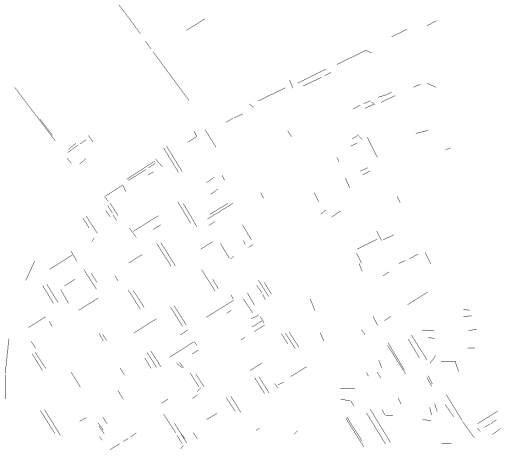


EDlines



LSDlines

PPHT



CannyLines

AG3Lines | ABEDL

Figure 116: Line detection results comparison for London CBD (TI_{10})

TI_2 | GT

EDlines | LSDlines

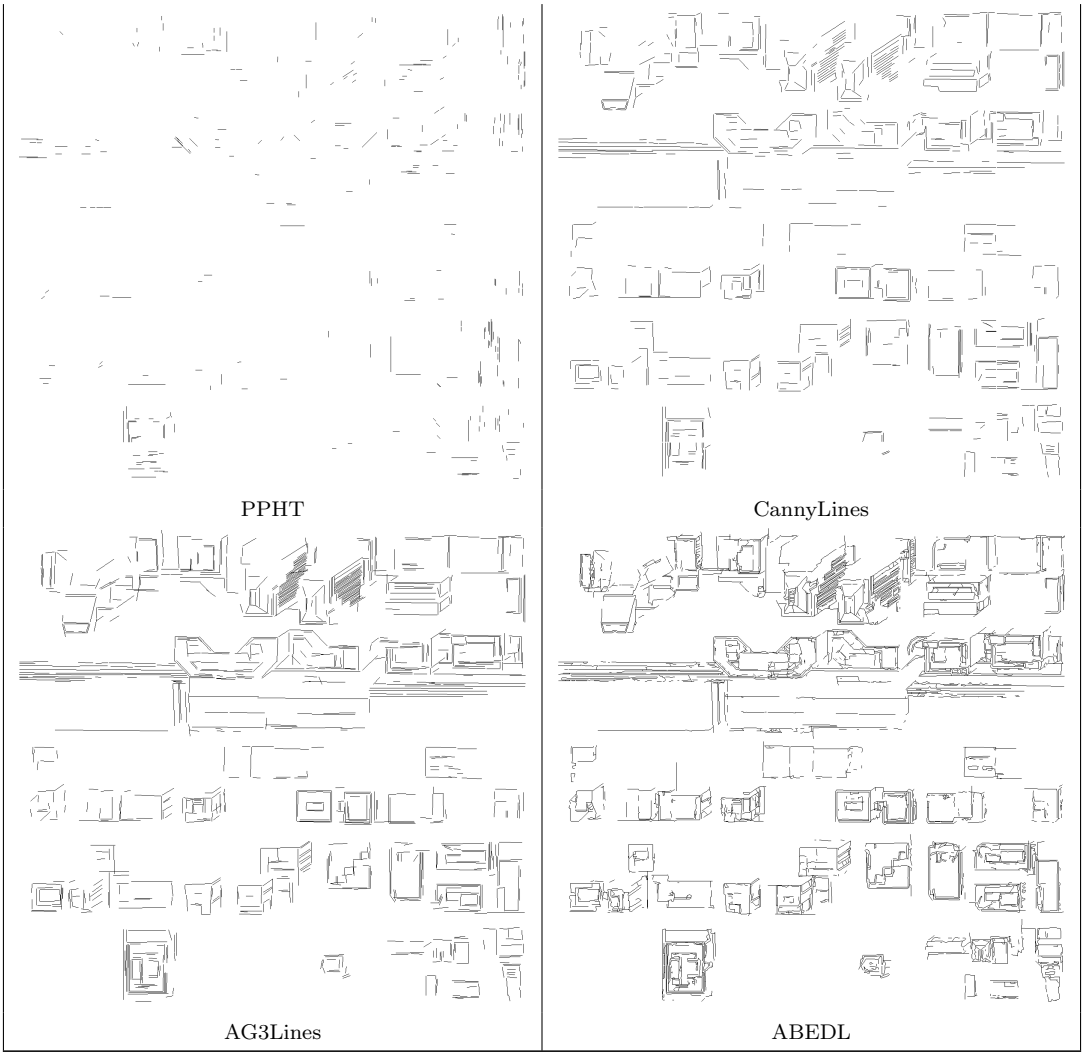
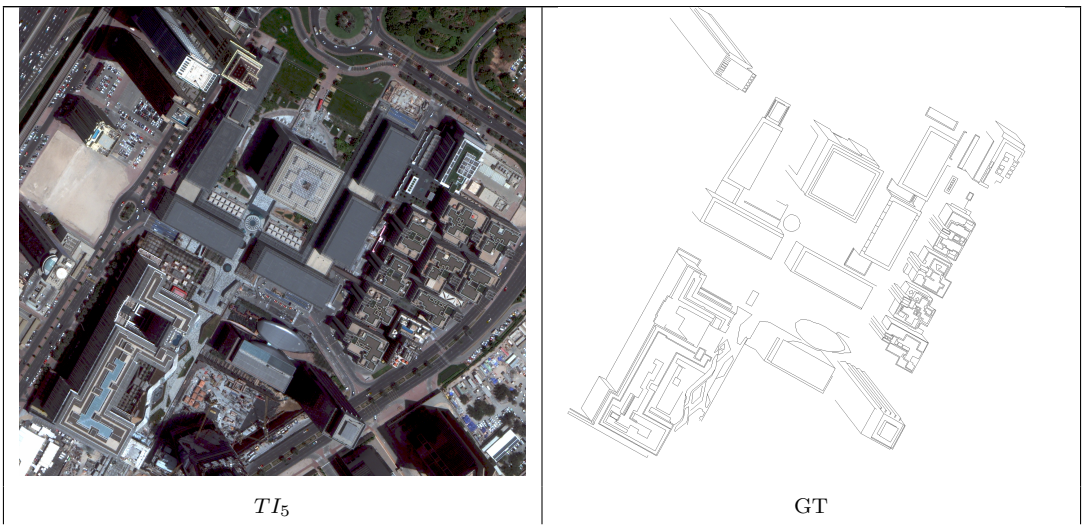


Figure 117: Line detection results comparison for Calgary CBD (TI_2). ABEDL, has denser and longer lines.



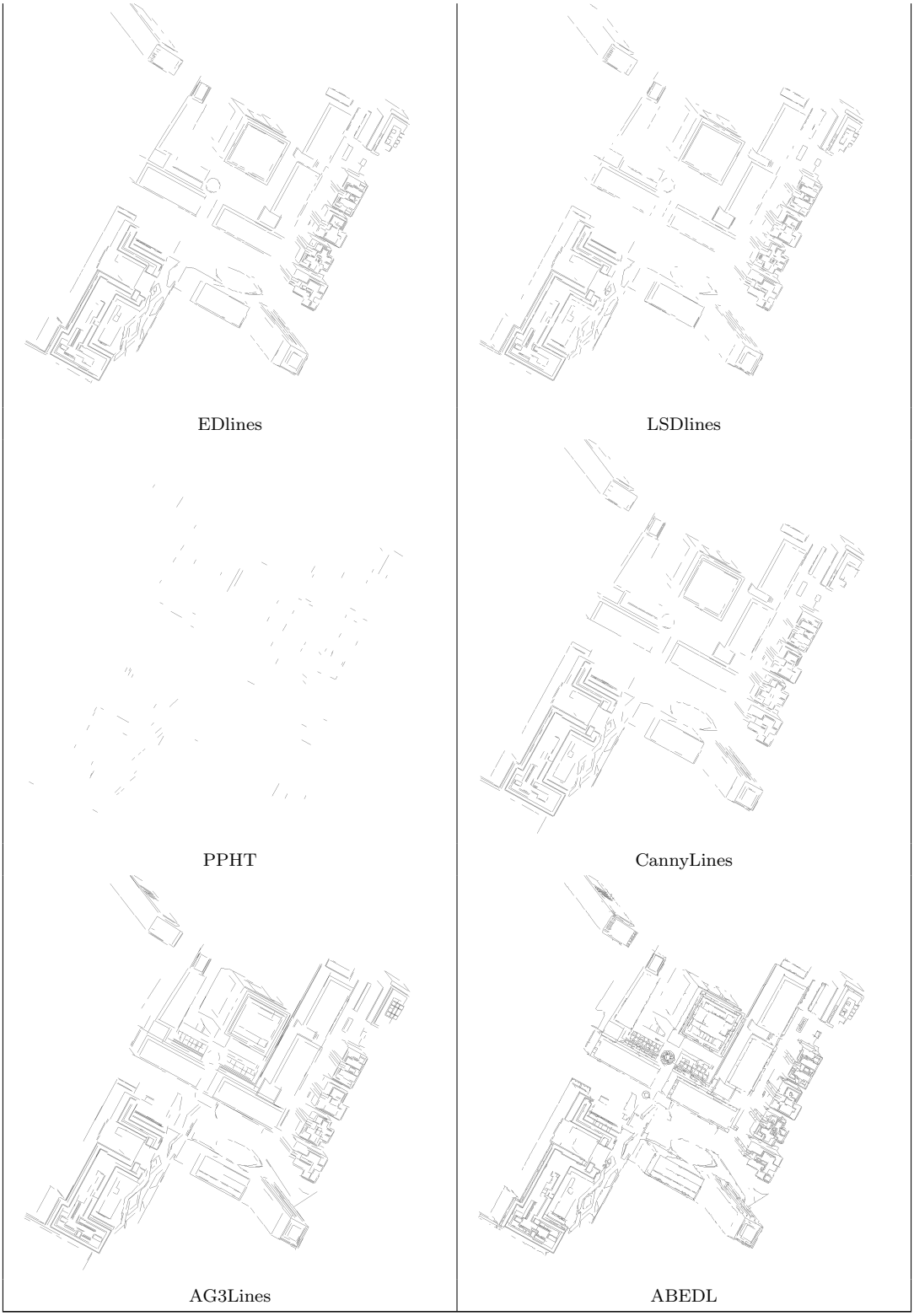
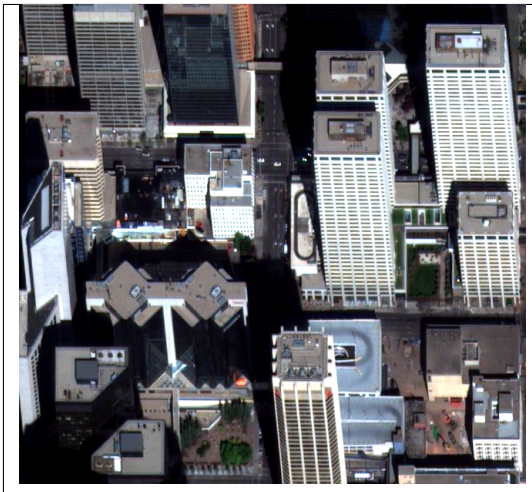
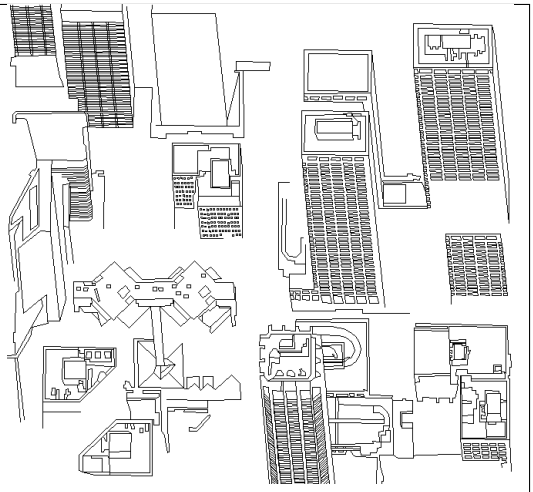


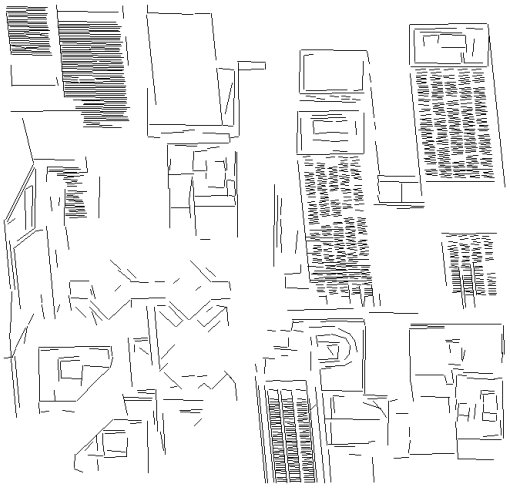
Figure 118: Line detection results comparison for Dubai CBD (TI_5)



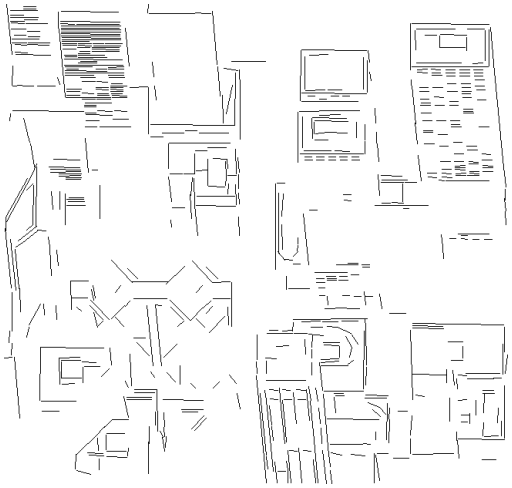
TI_3



GT

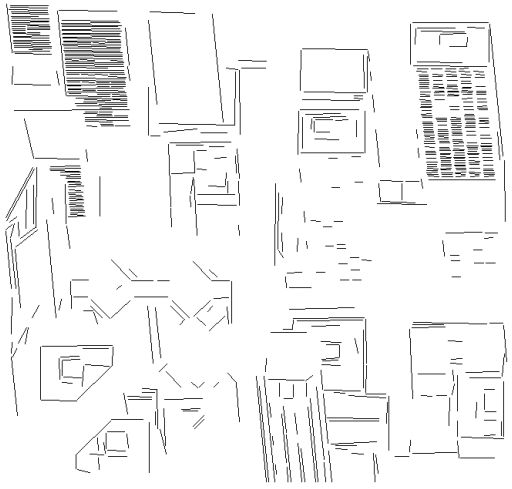


EDlines



LSDlines

PPHT



CannyLines

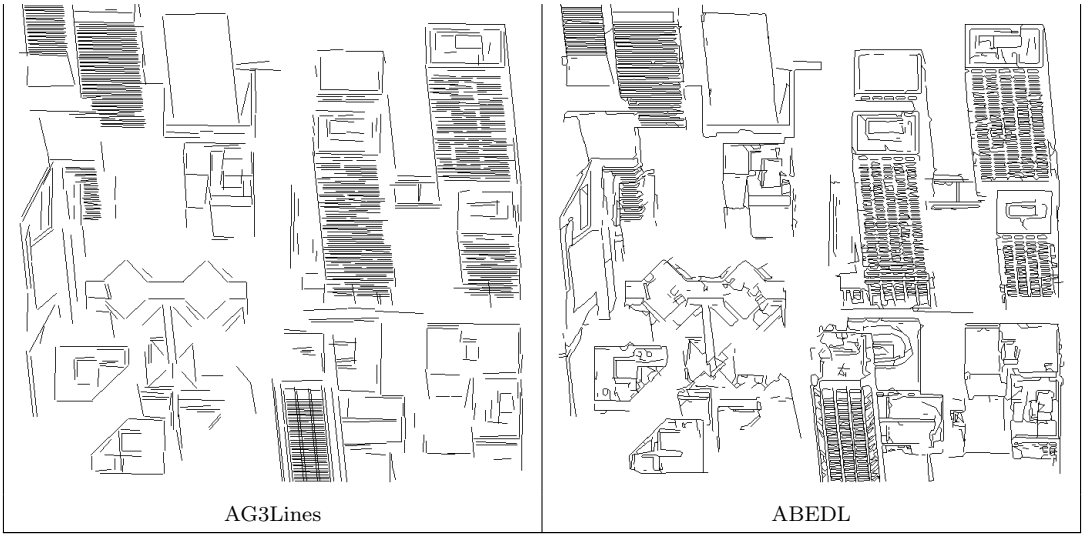
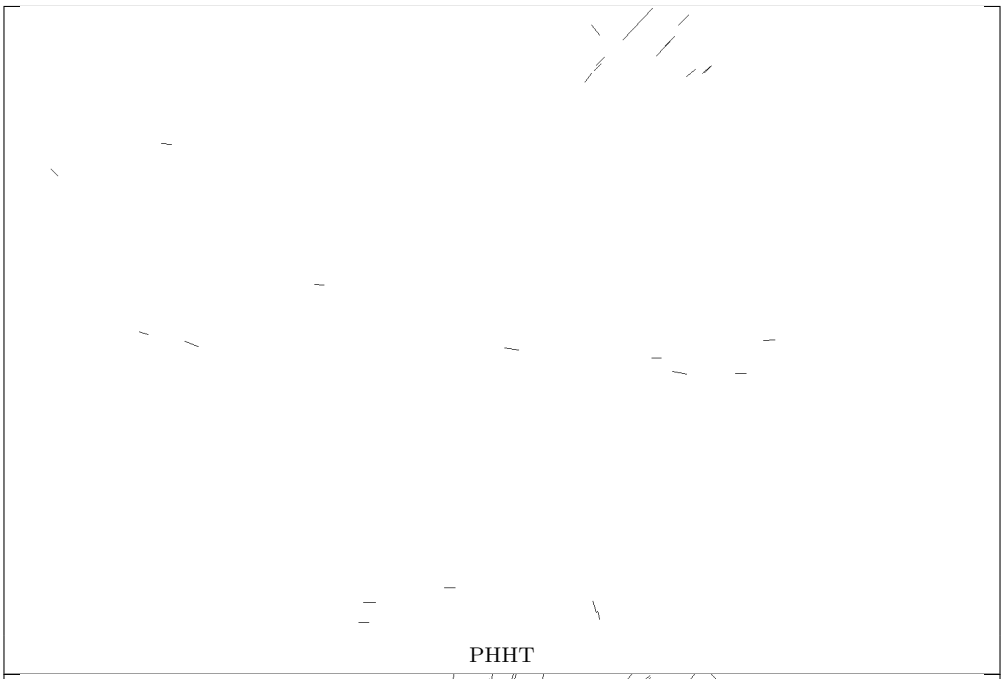


Figure 119: Line detection results comparison for Calgary CBD (TI_3)

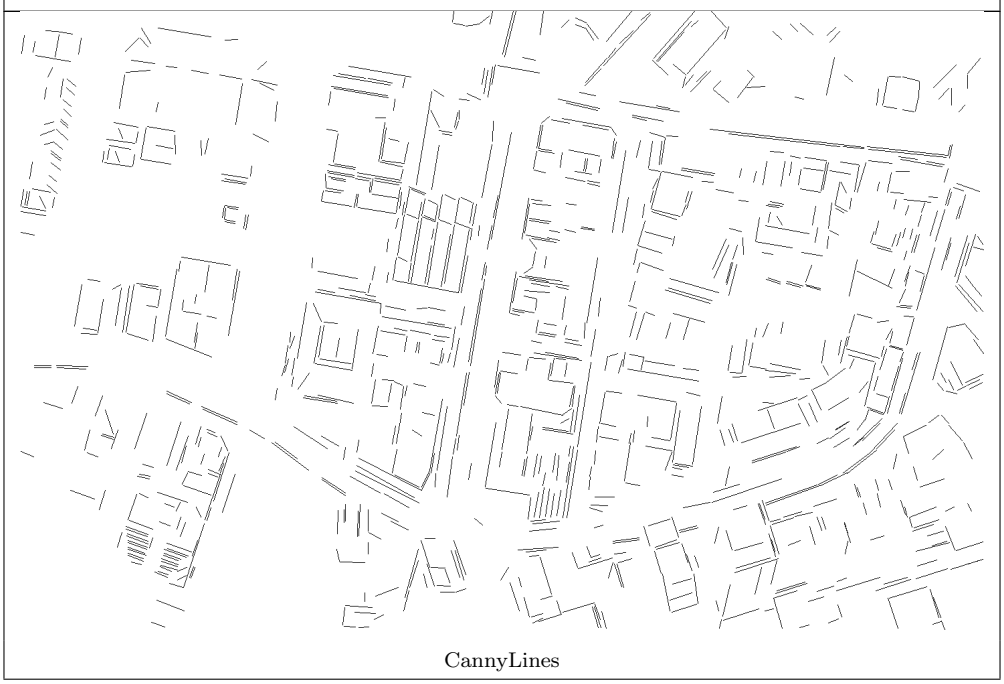
11 Appendix B

EDLines

LSD



PHHT



CannyLines

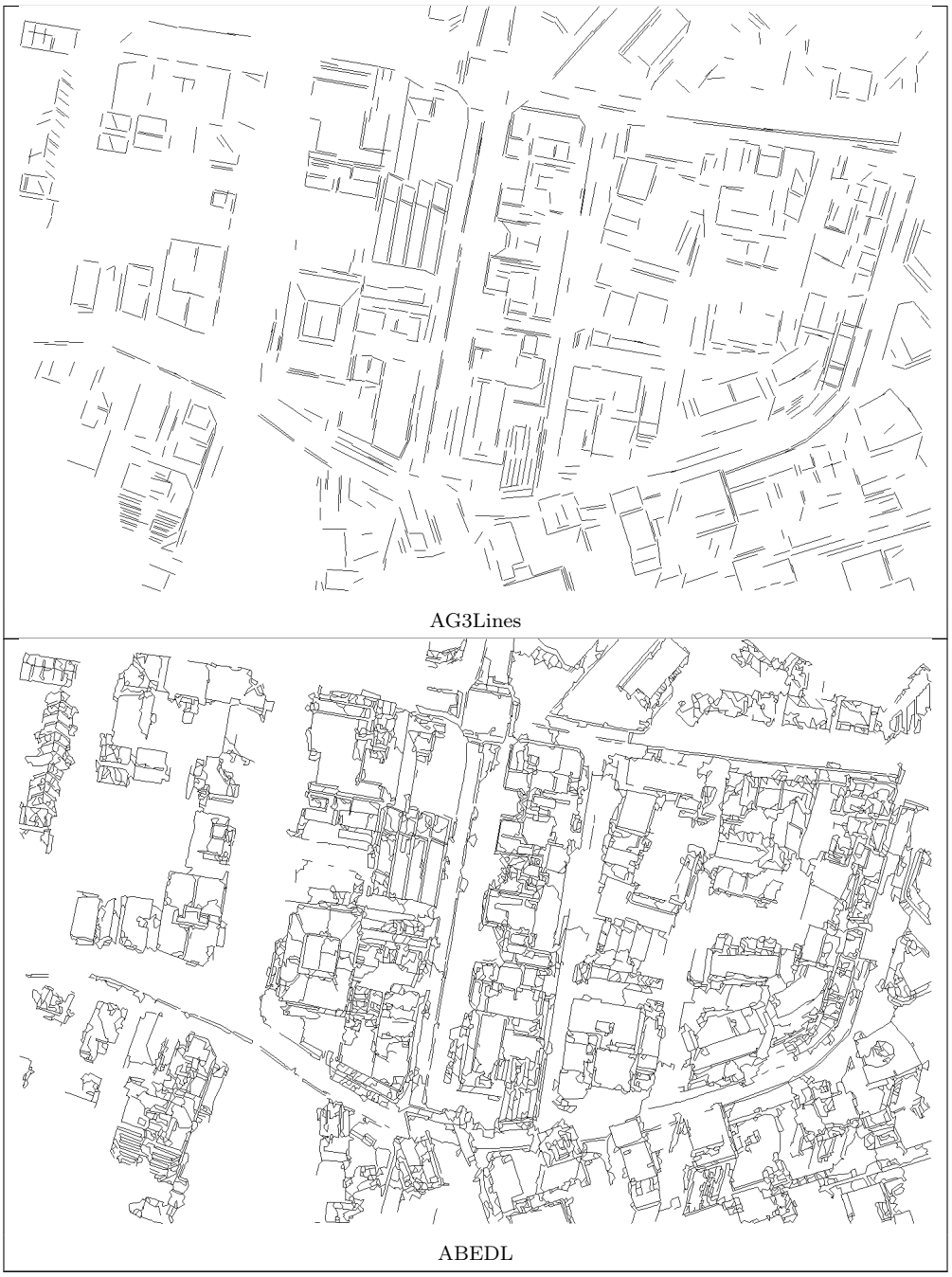
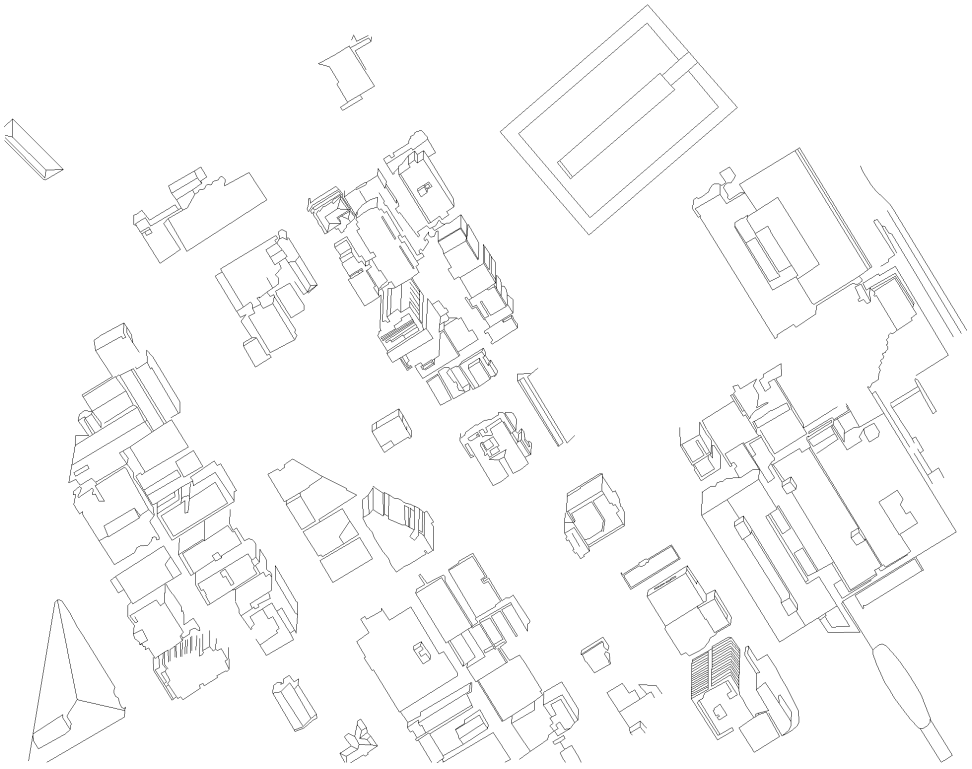


Figure 120: Boundary detection results comparison for Brisbane CBD (TI_1). ABEDL, has denser and longer lines. Links which connect valid lines have been retained to preserve object detail that improves wireframe results.

Image

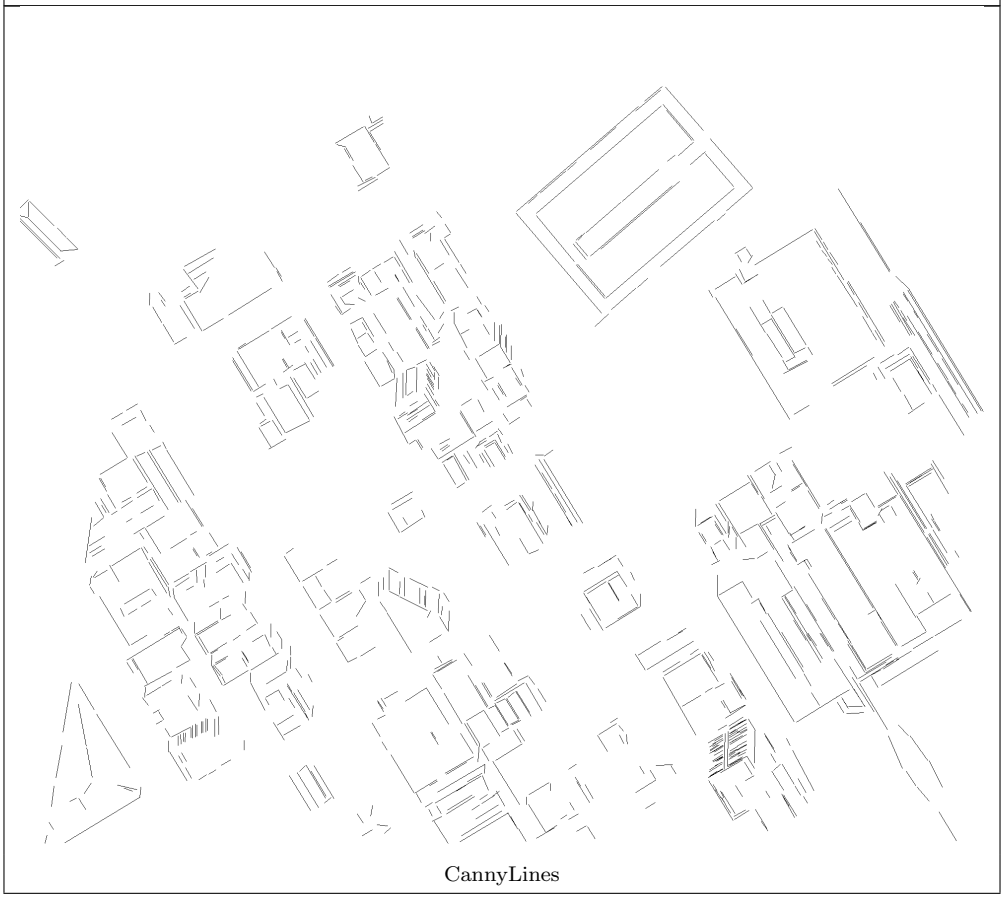


GT

EDLines

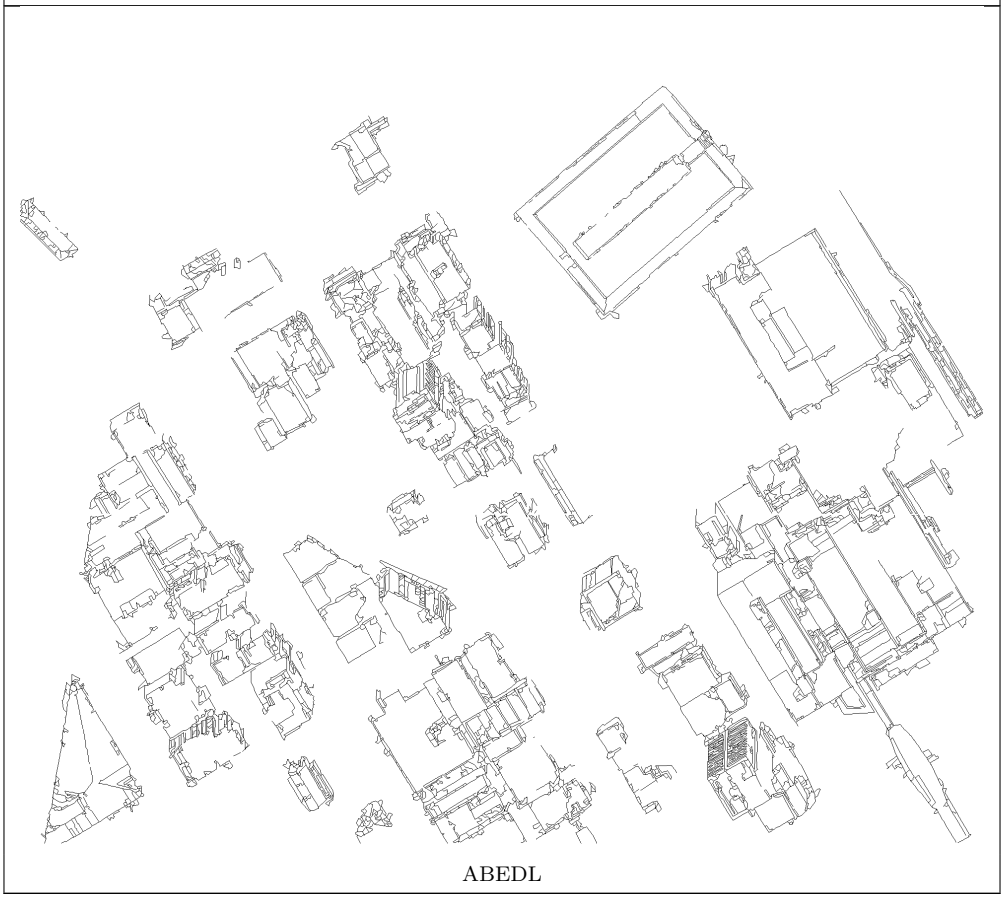
LSD

PHHT



CannyLines

AG3Lines

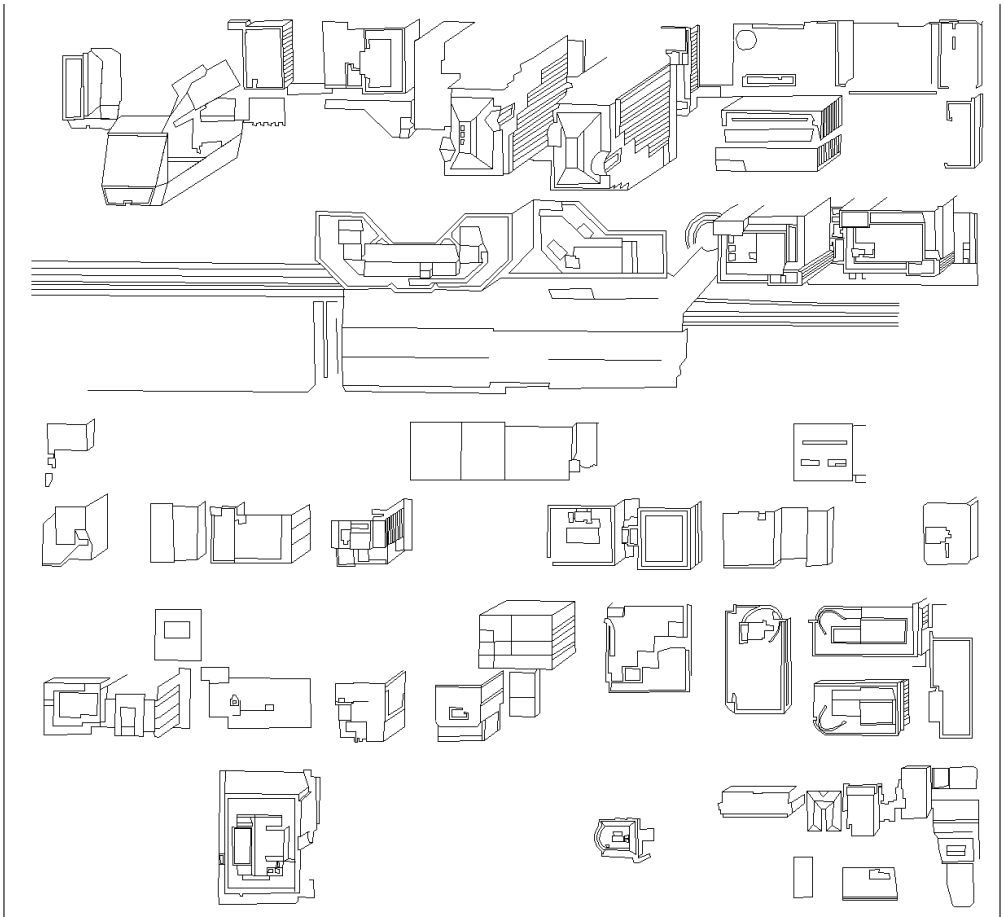


ABEDL

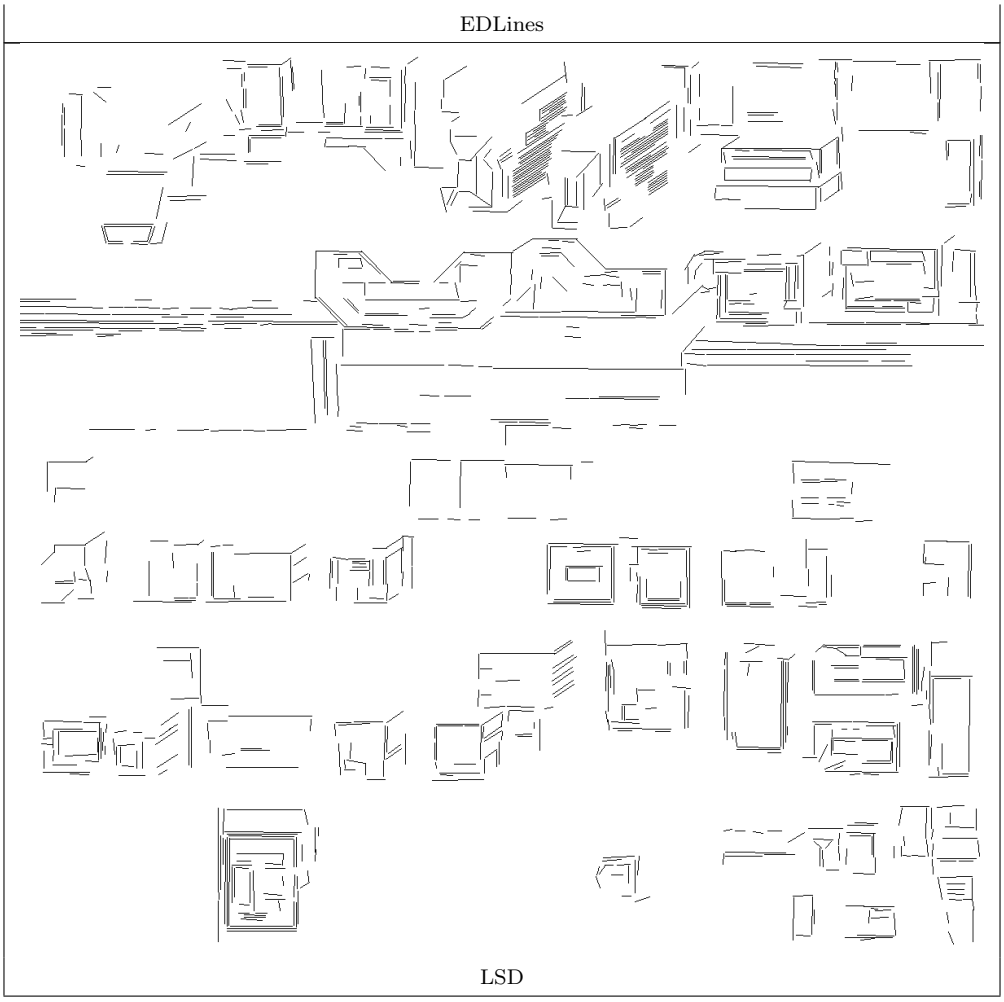
Figure 121: Boundary detection results comparison for Brisbane CBD (TI_4). ABEDL, has denser and longer lines.

[!htbp]

Image



GT


EDLines
LSD

AG3Lines

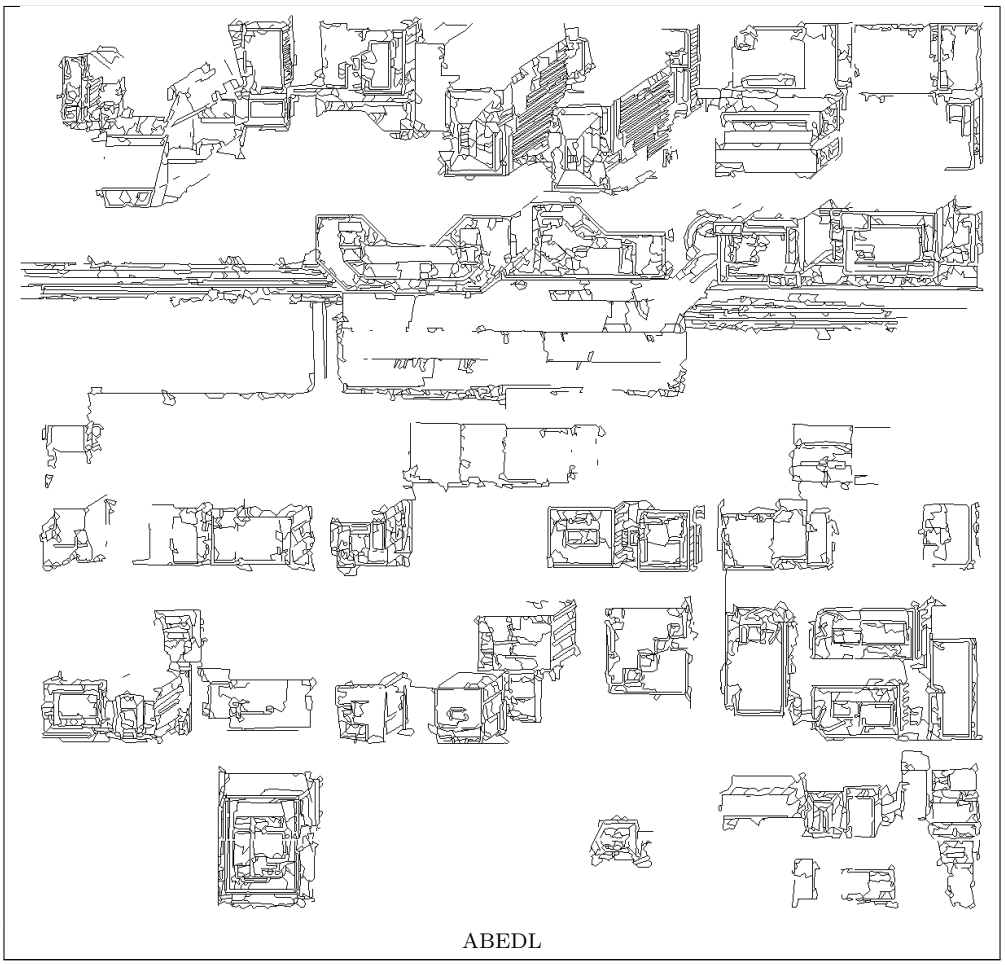


Figure 122: Boundary detection results comparison for Calgary CBD (TI_2). ABEDL, has denser and longer lines. Links which connect valid lines have been retained to preserve object detail that improves wireframe results.